2004－CSEC－27　(5)
2004／12／20

匿名データベースの速度改善と評価

An Implementation of Anonymous Database for High Performance

伊藤　聖吾*　　杉野　栄二　　片町　健太郎　　阿部　芳彦
岩手県立大学 ソフトウェア情報学部
*g231b002@edu.soft.iwate-pu.ac.jp**

概要

AnonySQLは，ユーザに対して匿名性を維持させたままで，データベースアクセスを実現するシステムである．実際の応用例として，交通事業者間における事故情報共有システムが存在している．現在，匿名性を必要とするシステムへの要求が存在しており，その実現に AnonySQLは有用であると考える．本研究では，AnonySQLの汎用利用を目的として，応答時間の改善と評価を行った．

1　はじめに

匿名でのデータベースアクセスを実現する AnonySQLは，事故情報共有システムを構築する目的で開発された [1]．本研究では，AnonySQLの性能を分析評価し，処理速度の改善を図り，汎用利用に耐えうるようにすることを目的とする．

匿名性に関する研究として，北川らによる講義評価の研究 [2] と，藤村らによる匿名情報提供方式の提案 [3] がある．北川らは，学生の講義評価に対する精度を向上させるため，匿名で回答を集めるためのプロトコルを提案し，システムを試作した．彼らの試作システムは講義評価に特化したものであり，AnonySQLのような汎用利用は難しい．藤村らは，内部告発のための匿名情報提供方式を提案している．彼らの方式は，告発者に対し社会的な不利益が発生することを防止することを目的とし，そのために匿名での情報提供手段を提供する．内部告発では，複数回の情報提供に対し，情報提供者が同一人物であることを特定できなければならない．このようなシステムも，AnonySQLを利用することで容易に構築することが可能であると考えられる．

2　AnonySQL

AnonySQLは，クライアントと認証サーバとデータサーバ，匿名通信路 Gunshuから構成されている．Gunshuは HTTP 通信を匿名化する Crowds[4] を TCP 接続で匿名化できるように改良したものである．認証サーバは，クライアントに対して認証を行い，データベースへのアクセスを許可する．データサーバはアクセス制御部と DBMS（DataBase Management System）から構成される．アクセス制御部は，匿名ユーザのアクセス権をもとに，データベースレコードへのアクセス制御を行い，ODBC（Open DataBase Connectivity）経由で DBMSへアクセスする．ODBCを利用するため，AnonySQLでは特定の DBMSに依存せずに動作することが可能である．現実装では DBMSとして Microsoft SQL Serverを利用する．

INSERT処理を例にして，AnonySQLにおけるクライアント認証と DBMSアクセス手順を示す．図 1は，その概念図であり，図中の番号は以下に示す手順と一致している．

(1) クライアントは認証サーバにアクセスし，データベースアクセスの認証を受ける．認証されると，認証サーバからクライアントに対しリクエスト証明書が発行される．
(2) クライアントは，匿名通信路を経由してデータサーバに INSERT処理を依頼する．その際にリクエスト証明書を添付する．
(3) データサーバのアクセス制御部は，アクセス権を持った匿名クライアントからの依頼であることを確認し，INSERT処理を行う．
(4) データサーバのアクセス制御部はレコード所有者証明書を生成してクライアントに返す．レコード所有者証明書は，レコードの正当な所有者であることを示すものであり，後でクライアントが更新，削除などの処理を行う際に利用される．
(5) クライアントは，受け取ったレコード所有者証明書をレコード所有者証明書ストアと呼ぶデータベースに格納する．

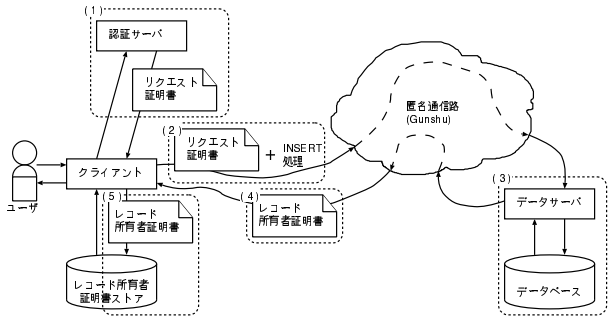

図 1: **AnonySQL** における **INSERT** 処理

3 AnonySQLの性能測定

AnonySQL と SQL Server に対し，処理対象レコード数を 1 件に限定して SELECT,INSERT,UPDATE の実行時間を測定した結果を図 **2** に示す．AnonySQL ではユーザに対して匿名性を提供するための処理が必要であり，SQL Server と比較して処理時間がその分だけ増加している．特に，UPDATE 処理時間は SELECT,INSERT と比べても大きい．

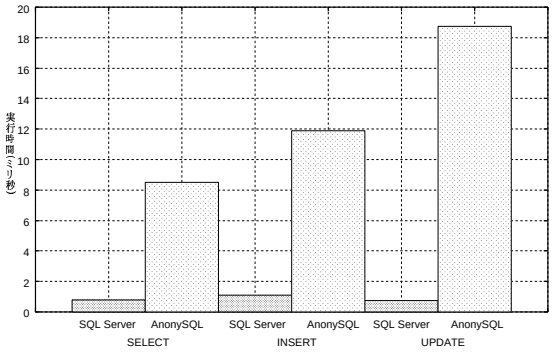

図 2: **SQL Server** と **AnonySQL** の性能比較

次に，対象となるレコード数を増加させて，UPDATE 処理を行った場合の性能比較結果を図 **3** に示す．SQL Server は，更新するレコード数を増加させても UPDATE 処理時間が増加しないのに対し，AnonySQL では UPDATE 件数に比例して処理時間が増加することが分かる．

AnonySQL を用いた最初の応用例である事故情報共有システムでは，既に登録してある事故情報に対する SELECT 処理が主な利用形態であるため，UPDATE 処理に時間がかかることはあまり問題とはされなかった．しかし，一度に大量の UPDATE 処理を行うようなシステムに AnonySQL を利用する場合，図 **3** で示されるような処理時間は大いに問題となる．そのため，AnonySQL を汎用的に利用するには，まず UPDATE

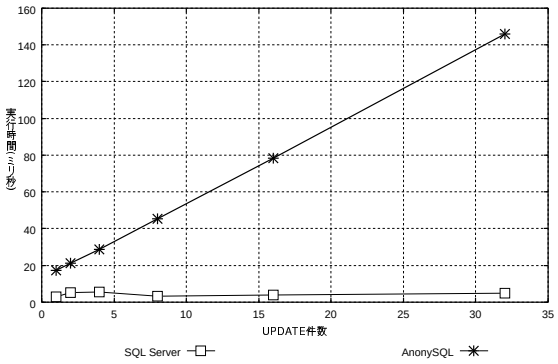

図 3: **AnonySQL** における **UPDATE** 処理性能

処理の高速化が必要である．

4 UPDATE処理の分析

図 **4** にクライアントとデータサーバ間における UPDATE 処理シーケンスを示す．図 3 の★部分で，UPDATE 対象となるレコードに対し，レコード所有者証明書が必要になる．これはクライアントが正当なレコード所有者であるかデータサーバ側で確認するためである．

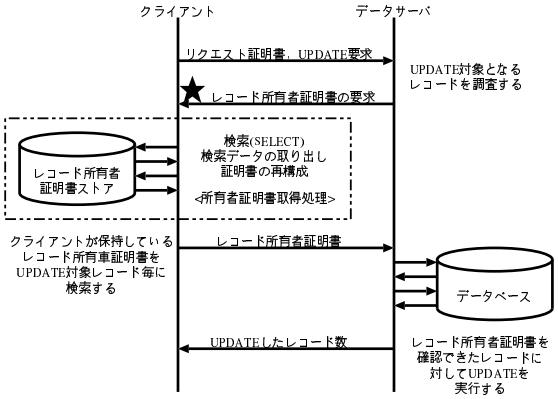

図 4: **AnonySQL** における **UPDATE** 処理シーケンス

データサーバからクライアントに対してレコード所有者証明書が要求されると，クライアント側では所有者証明書取得処理が発生する．この処理はレコード所有者証明書の検索，取得データの取り出し，証明書の再構成という流れで行われる．レコード所有者証明書の検索は，クライアント側のレコード所有者証明書ストアと呼ばれる DBMS に対して行われる．最後に，取り出された証明書はデータサーバへの送信形式に再構成される．この所有者証明書取得処理は UPDATE 対

象となるレコード毎に行われるため，レコード数に比例した処理時間がかかる．

所有者証明書取得処理時間を計測し，UPDATE処理全体に占める割合を表したものが図 5 である．UPDATE件数が 4 件を超えたところで，所有者証明書取得時間が全体の処理時間の 50%以上を占める．したがって，所有者証明書取得処理の改善が，全体の速度向上に最も有効であると考える．

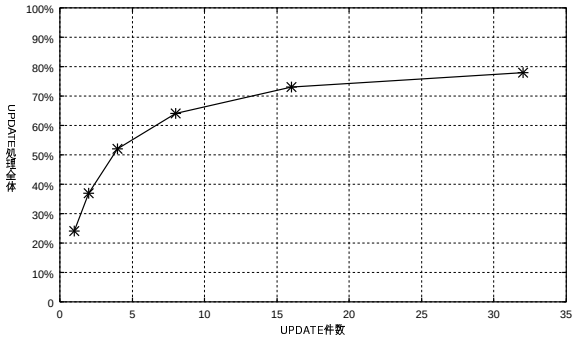

図 5: **AnonySQL の UPDATE 処理における所有者証明書取得時間の占める割合**

5 速度改善と評価

AnonySQLの所有者証明書取得処理は，図 6 のように，レコード毎に検索を行い，データの取出しと証明書の再構成まで行っている．

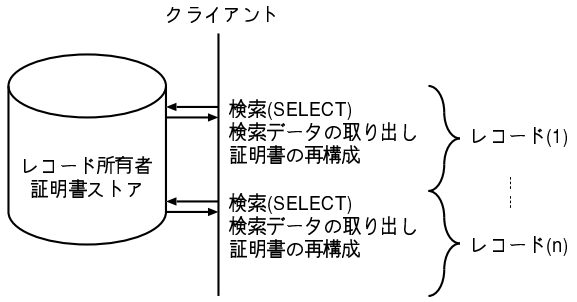

図 6: **既存の AnonySQL における所有者証明書取得処理シーケンス**

これに対し，検索処理を一括して行うことにより，DBMSに対する検索時間が減少すると考え，所有者証明書取得処理を図 7 のように改善した．改善を施したAnonySQLは，UPDATE件数が 32 件の場合で改良前の約 62%の処理時間になった．このときの所有者証明書取得処理の実行時間の内訳は図 8 のようになる．検索処理は予想通り減少したが，検索データの取り出しは時間はやや増加している．証明書の再構成処理は，検索してきた証明書数分だけ行われる．この部分には改善を加えていないため，改良版でもオリジナルと同じ時間で実行される．

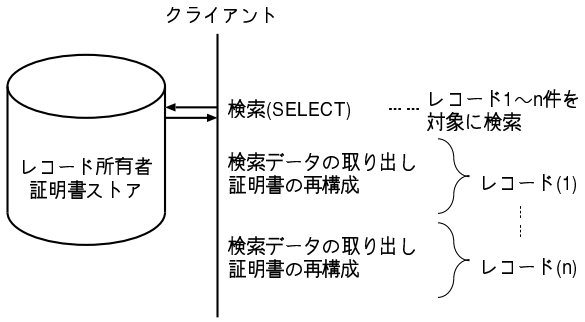

図 7: **改善を加えた AnonySQL における所有者証明書取得処理シーケンス**

北川らの講義評価アンケート [2] では，一つの講義科目に対して一度回答したアンケートは変更できない．これに対し，AnonySQLを用いた講義評価アンケートでは，UPDATE処理によって回答内容の更新が可能である．アンケート項目毎にレコードを構成するものとすれば，1科目の更新は複数レコードに対するUPDATE処理として実行される．複数の講義に対し一括して更新を行った場合は，UPDATE件数がさらに増大する．したがって本研究の速度改善は，このようなアンケートシステムに対して大いに有効である．

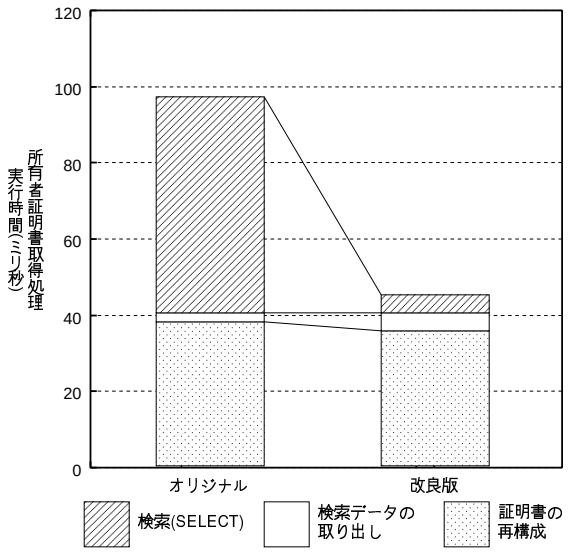

図 8: **改善を加えた所有者証明書取得処理の実行時間**

6 まとめと今後の課題

本研究は，匿名でのデータベースアクセスを実現する AnonySQL を，事故情報共有システム以外の事例にも適用可能にすることを目的とする．最も処理時間がかかる UPDATE の速度改善を行ったところ，件数が 32 件の時に改良前の 62%の処理時間になった．

今後は匿名性を確保した講義評価アンケートシステムや匿名情報提供システムの実装等，AnonySQL を利用したアプリケーションを構築する計画である．

参考文献

[1] 対馬伸行,「匿名データベースによる交通事故情報の企業間共有」，岩手県立大学 ソフトウェア情報学研究科 博士前期課程論文，2003

[2] 北川隆，岡博文，楫勇一,「大学における講義評価のための匿名アンケートプロトコルとその試作」，情報処理学会論文誌，Vol.44，No.9，pp.2353-2362(Sep.2003)

[3] 藤村明子，鈴木幸太郎，森田光,「内部告発制度の法制化を考慮した匿名情報提供方式」，Information Netwrko Law Review，Vol.2，pp.29-43(2003)

[4] M. K. Reiter and A. D. Rubin，“Anonymity leves company: Anonymous Web transactions with Crowds”，Comm. of the ACM 42(2) pp. 32-38(1999)

[5] O. Berthold，H. Federrath，and M. Kohntopp，“Project anonymity and unobservability in the internet.” In Computers Freedom and Privacy Conference 2000(CFP 2000) Workshop on Freedom and Privacy by Design，April 2000.

[6] Michael G. Reed，Paul F. Syverson，and David M. Goldschlag，Naval Research Laboratory，“Anonymous Connections and Onion Routing”，IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection，1998

[7] Matthew Wright，Micah Adler，Brian N.Levine，Clay Shields，“An Analysis of the Degration of Anonymous Protocols”，Network and Distributed System Security Symposium(NDSS) Conference Proceedings: 2002，`http://www.isoc.org/`(Last Visit 2004/06/30)