

A Comparative Study of Query Concurrency Control Techniques in Relational Database Management Systems

BIDUAYA MUKANA Clément

Licencié en Sciences Informatiques et Assistant à l'Université Notre-Dame du Kasayi (U.KA.), Kasai Central, Kananga (RDC).

Clementbiduaya27@gmail.com

Abstract: Concurrency control is a crucial factor determining the performance optimization and reliability of Relational Database Management Systems (RDBMS) [source: 1]. This paper presents a rigorous empirical comparative evaluation between two major isolation algorithms implemented within the MySQL 8.0 InnoDB storage engine: Multi-Version Concurrency Control (MVCC) and Strict Two-Phase Locking (S2PL). Using the Sysbench benchmarking tool, Online Transaction Processing (OLTP) workloads were simulated on an initial dataset of 100,000 records across moderate (10 concurrent threads) and intensive (50 concurrent threads) user environments. Experimental results demonstrate the operational superiority and high scalability of the MVCC protocol, achieving a throughput of 29.48 transactions per second (TPS) under 50 threads with zero deadlocks. Conversely, the S2PL protocol suffered an exponential performance degradation under high data contention, causing critical system saturation marked by 2,005 deadlocks within 30 seconds and a doubled average latency of 3,167.06 ms. This study provides clear engineering and managerial guidelines for information systems, recommending the strict adoption of MVCC for high-concurrency OLTP applications while restricting S2PL to critical financial transactions requiring absolute serializability.

Keywords—RDBMS; MySQL; InnoDB ; concurrency control ; MVCC ; S2PL ; deadlocks ; OLTP performance.

Résumé : Le contrôle de concurrence est un paramètre déterminant pour l'optimisation des performances et la fiabilité des Systèmes de Gestion de Bases de données Relationnelles (SGBDR). C'est article propose une évaluation comparative empirique entre deux algorithmes majeurs d'isolation implémentés au sein du moteur InnoDB de Mysql 8.0 : le contrôle de Concurrence Multiversion (MVCC) et le Verrouillage Biphase Strict (S2PL). A l'aide de l'outil de benchmarking Sysbench, des charges de travail de type OLTP (Online Transaction Processing) ont été simulées sur un jeu de données initial de 100000 enregistrements, e, confrontant des environnements à 10 et 50 utilisateurs concurrents. Les résultats expérimentaux démontrent la supériorité opérationnelle et la forte élasticité du protocole MVCC, qui atteint un débit de 29.48 transactions par seconde (GTPS) sous 50 threads sans générer le moindre interblocage (O deadlock). A l'inverse, le protocole S2PL subit une dégradation exponentielle sous forte contention, provoquant une saturation critique marquée par 2005 interblocages en 30 secondes et un doublement de la latence moyenne (3 167.06). Cette étude fournit des directives managériales précises pour l'ingénierie des systèmes d'information, préconisant l'adoption rigoureuse du MVCC pour les applications OLTP à haute fréquentation et restreignant l'usage du S2PL aux transactions financières critiques, exigeant une sérialisabilité absolue.

Mots-clés—SGBDR ; MySQL ; InnoDB ; contrôle de concurrence ; MVCC ; S2PL ; inter blocages ; performances OLTP.

1. INTRODUCTION

L'optimisation des performances dans les Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) pose ses bases fondamentales sur l'efficacité de leurs mécanismes de contrôle de concurrence. Dans les contextes à haute densité transactionnelle, le compromis entre l'isolation stricte des transactions (propriétés ACID) et le débit global du système (throughput) demeure un défi d'ingénierie majeur [1]. Cette étude se concentre sur le SGBDR MySQL (avec le moteur InnoDB), l'un des SGBDR open-source les plus déployés à l'échelle mondiale. Bien qu'InnoDB utilise par défaut un mécanisme hybride, il est crucial de comprendre comment ses algorithmes sous-jacents se comportent sous stress. Cet article propose une évaluation comparative rigoureuse entre le contrôle de concurrence multiversion (MVCC), favorisant

l'optimisme en lecture, et le verrouillage biphase strict (S2PL), garantissant la sérialisabilité au détriment de la concurrence [2].

2. REVUE DE LITTÉRATURE ET ALGORITHMES

2.1 Le protocole Biphase Strict (Strict 2PL)

Le protocole 2PL est le protocole historique fondamental garantissant la sérialisabilité et l'absence d'annulations. Dans sa variante stricte (S2PL), telle qu'implémentée pour les requêtes FOR UPDATE ou le niveau d'isolation *Serializable* dans MySQL, tous les verrous exclusifs (écriture) acquis par une transaction sont maintenus jusqu'à la validation de cette dernière (Commit) ou à son annulation (Rollback) [3]. Si cette approche prévient le phénomène de lecture sale (dirty read) et les mises à jour

perdues, elle augmente considérablement la probabilité d'interblocages (deadlocks) et dégrade la latence lors de fortes contentions sur les mêmes tuples.

2.2 Le Contrôle de Concurrency Multiversion (MVCC)

Pour résoudre les limitations du 2PL, InnoDB intègre le MVCC pour ses niveaux d'isolation *Repeatable Read* (par défaut) et *Read Committed*. Le MVCC permet à une transaction de lire un instantané (snapshot) des données sans acquérir de verrous bloquants [4]. Chaque ligne possède des métadonnées précisant sa version. Ainsi, les opérations de lecture sur les données ne bloquent pas les opérations d'écriture, et vice versa, ce qui augmente drastiquement le débit transactionnel dans les charges de travail orientées lecture (Read-Heavy workloads) [5].

3. MÉTHODOLOGIE EXPÉRIMENTALE

3.1 Environnement matériel et logiciel

Pour garantir la reproductibilité des résultats, les tests ont été exécutés sur un environnement local dont les caractéristiques matérielles sont les suivantes : un processeur AMD Athlon Silver 3050U (2.30 GHz), équipé de 4 Go de mémoire vive (RAM). Le système d'exploitation hôte est un environnement Windows 10 Professionnel (architecture 64 bits). Le SGBDR évalué est MySQL 8.0, configuré avec le moteur de stockage transactionnel InnoDB.

3.2 Outil d'évaluation et de charge

Afin de simuler une charge transactionnelle représentative des environnements de production réels, l'outil de *benchmarking* Sysbench a été sélectionné. Cet utilitaire standardisé permet de générer des charges de travail synthétiques de type OLTP (*Online Transaction Processing*) de manière fortement concurrentielle (multi-threadée). Il est particulièrement adapté pour mesurer avec précision l'impact des algorithmes d'isolation sur le débit transactionnel global (Transactions Par Seconde - TPS) et sur la latence moyenne de réponse.

3.3 Scénarios de test et protocole

Le protocole expérimental s'articule autour de deux configurations distinctes du niveau d'isolation des transactions au sein du moteur InnoDB :

- **Premier scénario (Optimisation MVCC) :** Le système est configuré avec le niveau d'isolation par défaut *REPEATABLE READ*. Ce scénario évalue la capacité d'InnoDB à traiter des lectures concurrentes via des instantanés (snapshots) sans verrouiller les opérations d'écriture simultanées.
- **Deuxième scénario (Verrouillage S2PL) :** Le système est forcé d'utiliser un verrouillage strict, soit en modifiant le niveau d'isolation global à *SERIALIZABLE*, soit par l'injection massive de requêtes de type *SELECT*, *FOR*, *UPDATE*. L'objectif

est d'observer le comportement du système face à une haute contention où l'intégrité absolue des données prime sur la concurrence.

Pour initialiser l'environnement, une table de test contenant 100 000 enregistrements a été générée. Les deux scénarios ont ensuite été soumis à des charges progressives en augmentant le nombre d'utilisateurs virtuels concurrents 10, et 50 threads afin de mesurer la dégradation des performances et la fréquence d'apparition des interblocages (*deadlocks*).

4. RÉSULTATS ET DISCUSSION

4.1 Analyse comparative des performances

Les expériences menées via l'outil Sysbench ont permis d'évaluer le comportement et la capacité de mise à l'échelle (scalability) d'InnoDB sous deux paradigmes d'isolation distincts (MVCC vs S2PL). Pour observer l'impact de la contention, le protocole expérimental a soumis une table initiale de 100 000 enregistrements à deux niveaux de charge de travail : une charge modérée de 10 utilisateurs simultanés (threads), puis une charge intensive de 50 utilisateurs simultanés.

Les résultats bruts issus des bornes d'exécution de l'environnement Linux (WSL) sont intégrés sous forme de figures empiriques afin d'illustrer le comportement du SGBDR en temps réel.



Figure 1: Rapport d'exécution Sysbench pour le protocole S2PL sous une charge modérée de 10 threads concurrents (109 interblocages constatés)

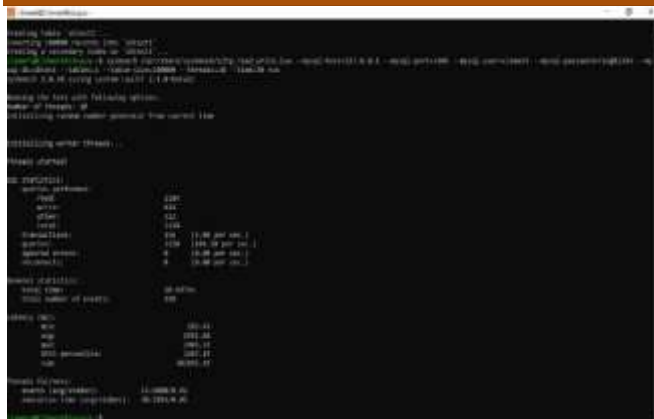


Figure 2 : Rapport d'exécution Sysbench pour le protocole MVCC sous une charge modérée de 10 threads concurrents.

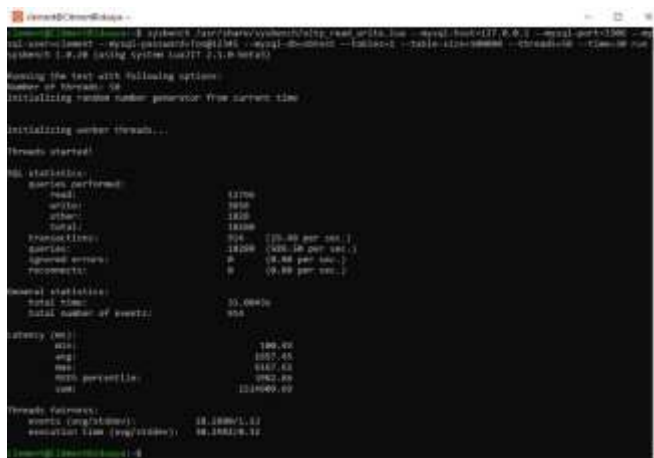


Figure 3: Rapport d'exécution Sysbench pour le protocole MVCC sous une charge intensive de 50 threads concurrents (0 interblocage, excellente adaptabilité).

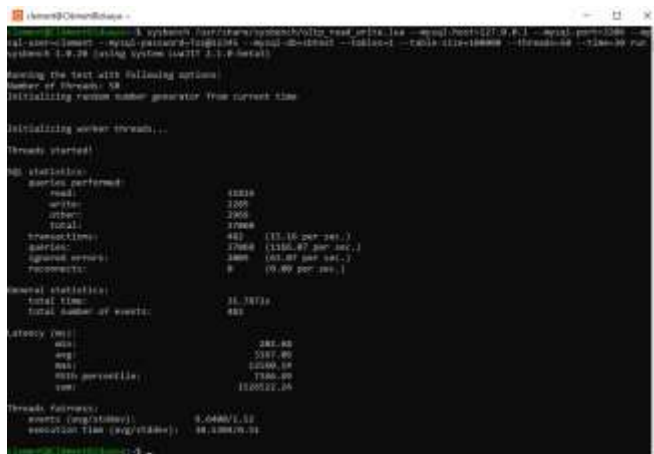


Figure 4: Rapport d'exécution Sysbench pour le protocole S2PL sous une charge intensive de 50 threads concurrents (2005 interblocages, effondrement transactionnel).

Transactions par seconde (TPS)	5.08	6.09
Latence moyenne (ms)	1941.64	1626.65
Latence maximale (ms)	3903.33	6799.45
Erreurs (Rejets / Deadlocks)	0	109
Requêtes totales exécutées	3120	5379

Tableau 1 : Représentation des données test sur 10 utilisateurs.

Métrique	Scénario1 : MVCC	Scénario2 : S2PL
Transactions par seconde (TPS)	29.48	15.16
Latence moyenne (ms)	1657.45	3167.06
Latence maximale (ms)	6167.61	12560.14
Erreurs (Rejets /Deadlocks)	0	2005
Requêtes totales exécutées	18280	37068

Tableau 2 : Représentation des données test sur 50 utilisateurs.

4.2. Discussion sur l'effondrement sous contention et recommandations

La comparaison directe entre les charges à 10 et 50 threads met en lumière des comportements radicalement opposés selon l'algorithme sous-jacent :

- La suprématie de l'élasticité du MVCC : Le mécanisme de contrôle par multiversion (Repeatable Read) prouve ici son excellente capacité de passage à l'échelle. Lors de la multiplication par 5 de la charge (de 10 à 50 threads), le débit transactionnel a été multiplié par près de 6 (passant de 5.08 à 29.48 TPS), tandis que la latence moyenne s'est optimisée (1657.45 ms contre 1941.64 ms). Le point le plus critique réside dans l'absence totale d'erreur (0 deadlock). Grâce aux lectures par instantanés (snapshots), les transactions de lecture n'ont imposé aucun verrou aux écritures, permettant aux ressources CPU de s'exprimer pleinement sans gaspillage d'opérations.
- L'effondrement exponentiel du S2PL sous haute contention : À l'inverse, l'application du verrouillage biphasé strict via le niveau Serializable révèle une vulnérabilité architecturale majeure sous forte concurrence. Si à 10 threads le système montrait déjà des signes de faiblesse (109 interblocages), le passage à 50 threads a provoqué une saturation critique marquée par 2 005 erreurs (deadlocks) en 30 secondes. Dans cette configuration, la latence moyenne a doublé (3167.06 ms) et certains utilisateurs virtuels ont subi des temps d'attente extrêmes dépassant 12 secondes. Le débit de 15.16 TPS observé ne reflète plus un système sain, mais une surcharge de traitements avortés où le SGBDR passe plus de temps à annuler et retransmettre des requêtes

Métrique	Scénario1 : MVCC	Scénario2 : S2PL
----------	------------------	------------------

bloquées (rollback overhead) qu'à valider les modifications.

4.2 Synthèse managériale pour l'ingénierie des Systèmes d'Informations

Ces résultats empiriques apportent des directives claires pour les concepteurs de Systèmes d'Information :

- Le protocole MVCC doit être rigoureusement privilégié pour les applications OLTP (Online Transaction Processing) modernes telles que le commerce électronique, les réseaux sociaux, ou les ERP, où la concurrence d'accès est élevée et la tolérance aux pannes/ralentissements est faible.
- Le protocole S2PL ne doit jamais être appliqué globalement sur une base de données à forte fréquentation. Son usage doit être chirurgicalement limité à des transactions financières critiques ou des opérations comptables isolées où l'exactitude mathématique et la prévention des lectures fantômes supplantent la nécessité de vitesse et d'accessibilité.

5. Conclusion

Cette étude empirique met en évidence des différences de performances majeures entre les protocoles de contrôle de concurrence MVCC et S2PL au sein du moteur InnoDB de MySQL 8.0.

L'analyse démontre clairement la supériorité opérationnelle et l'excellente élasticité du protocole MVCC pour la gestion des charges concurrentielles intenses au sein des SGRDR. Lors des tests simulant 50 utilisateurs simultanés, le MVCC a su maintenir un débit élevé de 29,48 transactions par seconde tout en évitant totalement les interblocages (0 deadlock). En revanche, le protocole de verrouillage biphasé strict (S2PL) s'est avéré très vulnérable sous une forte contention de données. Il a subi une saturation critique provoquant 2005 interblocages en seulement 30 secondes, ainsi qu'un doublement de sa latence moyenne qui a atteint 3167,06 ms.

D'un point de vue managérial et architectural, il est donc fortement recommandé de privilégier l'adoption du MVCC pour les applications OLTP modernes à haute fréquentation, comme les plateformes de commerce électronique, les réseaux sociaux ou les systèmes ERP. L'utilisation du S2PL ne doit pas être appliquée de manière globale, mais plutôt réservée de manière ciblée aux transactions financières et opérations comptables critiques qui exigent une intégrité absolue et une stricte sérialisabilité, au détriment de la vitesse d'exécution.

6. Bibliographies

[1] Haerder, T., & Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys (CSUR)*.

[2] Bernstein, P. A., Hadzilacos, V., & Goodman, N. (1987). *Concurrency control and recovery in database systems*. Addison-Wesley.

[3] Eswaran, K. P., Gray, J. N., Lorie, R. A., & Traiger, I. L. (1976). The notions of consistency and predicate locks in a database system. *Communications of the ACM*, 19(11), 624-633.

[4] Wu, Y., Arulraj, J., Lin, J., Xian, R., & Pavlo, A. (2017). An Empirical Evaluation of In-Memory Multi-Version Concurrency Control. *Proceedings of the VLDB Endowment*, 10(12), 1985-1996.

[5] MySQL Documentation Team (2026). *MySQL 8.0 Reference Manual: InnoDB Multi-Versioning*. Oracle Corporation.