

### 1. 担当 PM

田中 邦裕（さくらインターネット株式会社 代表取締役社長）

### 2. クリエータ氏名

宮崎 祐介（慶應義塾大学 総合政策学部 総合政策学科）

中森 辰洋（慶應義塾大学 大学院政策・メディア研究科）

李 浩文（慶應義塾大学 大学院政策・メディア研究科）

### 3. 委託金支払額

2,880,000 円

### 4. テーマ名

高性能で耐故障な MySQL の開発

### 5. 関連 Web サイト

ソースコード：<https://github.com/mitou-Kamo/LinearDB-storage-engine>

### 6. テーマ概要

本プロジェクトは、既存の MySQL を基盤としつつ、その内部構造に踏み込んだ拡張および統合を行うことで、高性能かつ耐障害性を備えたデータベースシステムを実現することを目的としたものである。

データベースはアプリケーションの基盤として広く利用されており、その性能および可用性はシステム全体の品質を左右する。一方で、MySQL のような成熟したデータベースシステムは長年の改良により安定性と実績を持つ反面、その内部構造は複雑であり、設計上の制約や歴史的な実装の積み重ねによって、性能と可用性を同時に高い水準で実現することが難しいという側面がある。

具体的には、トランザクション処理におけるロックや同期の仕組み、ストレージエンジンの設計、レプリケーションの遅延や整合性の問題などが、性能と耐障害性のトレードオフを生む要因となっている。例えば、高い整合性を確保しようとすれば同期処理が増え性能が低下し、逆に性能を優先すれば障害時の整合性や復旧の複雑さが問題となる。このような構造的な課題は、外部ツールや設定変更だけでは解決が難しく、内部構造に踏み込んだ設計が必要となる。

本プロジェクトは、このような課題に対して、既存システムの外側で補完するのではなく、ストレージエンジンやトランザクション処理といった内部機構を拡張・統合することで、性能と可用性を同時に改善することを試みている。特に、LinearDB の統合を通じてデータ処理の効率化を図るとともに、レプリケーションやフェイルオーバー構成を組み合わせることで、実運用を前提とした耐障害性の向上を目指している。

また、本プロジェクトは単なる性能改善ではなく、「実際に運用可能なシステムとして成立させる」ことを重視している点に特徴がある。すなわち、コアとなるデータ処理部分の拡張、障害時の継続運用を可能とする可用性設計、そしてそれらを支える運用構成を統合し、全体として一つのシステムとして機能することを目標としている。

さらに、本プロジェクトは複数のクリエイターによるチーム開発として進められており、コア部分の実装、可用性設計、運用構成といった異なる領域を分担し、それぞれの成果を統合する形で進行している。このように役割を分担しながらも、最終的にはシステム全体として成立させることを前提としている点も、本プロジェクトの特徴である。

以上より、本プロジェクトは、既存の MySQL の制約に対して内部構造レベルでアプローチし、性能と耐障害性という本来トレードオフになりがちな要素を同時に扱いながら、実用的なデータベースシステムとして成立させることを目指した取り組みである。

## 7. 採択理由

本提案は、MySQL のトランザクション処理系を抜本的に再構築することにより、世界中で広く利用されている既存ソフトウェアの根本的な課題に挑むものである。性能と耐故障性を同時に追求する設計は、従来の延長線上にはない困難な実装課題を多数含み、まさに未踏性が感じられる。

提案者らは、MySQL ストレージエンジンの開発、PostgreSQL へのパッチ採用、トランザクション制御アルゴリズムの国際会議発表、さらに複数の受賞歴を通じて、基盤ソフトウェアレベルの実装力と専門性を実証している。単なる研究提案にとどまらず、OSS 公開、企業連携、学術発信といった出口戦略も具体的に構想されており、成果の社会的波及性も高く評価できる。

一方で、MySQL という大規模 OSS に対する実装は複雑で、互換性維持や保守性の観点から、すべての工程が計画通りに進むとは限らない。しかしながら、こうした制約を理解したうえで、独自の手法と分担体制によって挑戦しようとする姿勢は、未踏事業という場においてこそ支援すべきものであると判断し、採択した。

## 8. 開発目標

本プロジェクトの開発目標は、既存の MySQL を基盤としつつ、内部構造に踏み込んだ拡張および統合を通じて、高性能かつ高可用性を両立するデータベースシステムを実現することである。

前節で述べた通り、MySQL における主要な課題は、トランザクション処理、ロック機構、レプリケーション、障害復旧といった複数の要素が相互に影響し合うことにより、性能と耐障害性のトレードオフが発生する点にある。本プロジェクトでは、このトレードオフを前提とするのではなく、内部構造の拡張と再構成を通じて、両者を同時に改善することが目標として設定されている。

具体的には、以下の3つの観点が開発目標として位置づけられる。

第一に、データ処理性能の向上である。トランザクション処理における競合や同期のオーバーヘッドを低減し、高いスループットと低レイテンシを実現することが求められる。このため、既存の処理モデルに依存するのではなく、LineairDB の統合を通じて、より効率的なデータ処理機構を構築することが目標とされている。

第二に、耐障害性の確保である。システムの一部に障害が発生した場合でも、サービス全体としては継続可能な構成を実現し、ダウンタイムを最小化することが求められる。このため、レプリケーション構成の設計やフェイルオーバーの仕組みを含め、障害時の挙動を前提としたシステム設計が目標に含まれている。

第三に、実運用を前提としたシステムの成立である。単なるアルゴリズムやモジュールの開発にとどまらず、実際に利用可能なデータベースシステムとして動作することが求められる。このため、コアとなるデータ処理、可用性設計、運用構成を統合し、一体として機能するシステムを構築することが目標とされている。

また、本プロジェクトはチーム開発として進められているため、各要素を個別に最適化するだけでなく、それらをどのように統合するかという点も重要な目標となっている。すなわち、性能、可用性、運用性の各要素を個別に高めるのではなく、それらを同時に成立させるシステム設計を行うことが、本プロジェクトの本質的な開発目標である。

## 9. 進捗概要

本プロジェクトは、既存の MySQL に対して内部構造レベルでの拡張および統合を行うという方針のもとで開始された。開発初期においては、どのレイヤで問題にアプローチするか検討が行われている。一般的にはアプリケーション層やミドルウェア層での最適化、キャッシュ導入、負荷分散といった手法により性能や可用性を改善することが多いが、本プロジェクトではそれらでは本質的な課題の解決には至らないと判断され、ストレージエンジンやトランザクション処理といった内部機構に直接手を入れる方針が採用された。

この判断の背景には、MySQL における性能と可用性のトレードオフ構造がある。トランザクションの整合性を強く担保しようとするるとロックや同期処理が増加し性能が低下する一方、性能を優先すると障害時の整合性や復旧が複雑化する。この構造的な制約は設定や外部ツールだけでは解決できず、内部処理の設計そのものに踏み込む必要があると判断された。

この方針に基づき、開発は大きく 3 つの領域、すなわちコアとなるデータ処理部分、可用性設計、運用構成に分かれて進行した。

まずコア部分においては、LineairDB の統合が試みられた。この作業は単なるライブラリの導入ではなく、MySQL のストレージエンジンとして動作させるための実装を伴うものであり、MySQL 内部のデータ管理構造やトランザクション処理の前提を理解した上での設計が必要であった。実際には、トランザクションの開始・コミット・ロールバックのタイミングや、ログ処理との整合性、並行処理時の競合解決といった点で多くの不整合が発生し、その都度修正が行われている。

特に苦労が大きかったのは、既存の MySQL の処理モデルと LineairDB の処理モデルの間に存在する前提の違いである。MySQL は長年の実装の中で確立された動作仕様を持っており、内部の状態遷移やロックの扱いが複雑である。一方で LineairDB は異なる設計思想に基づいており、そのまま統合すると想定外の挙動が発生するケースが多かった。このため、単純な組み込みではなく、どの部分を適用し、どの部分を MySQL の既存機構に委ねるかといった切り分けが必要となった。

LineairDB の統合において特に困難であったのは、MySQL が InnoDB を前提として最適化されている点である。MySQL 本体には、ストレージエンジンの抽象化レイヤが存在する一方で、実際には InnoDB に特化したコードや前提が随所に組み込まれており、単に新しいストレージエンジンを実装するだけでは成立しない場面が多く発生した。

具体的には、トランザクション管理やロック処理、ログ処理、バッファ管理などの各所において、InnoDB の動作を前提とした処理が存在し、これらが LineairDB の設計と整合しないケースが確認された。その結果、ストレージエンジン側での実装だけでは対応できず、MySQL 本体のコードに対しても修正を加える必要が生じている。

この点は、本プロジェクトの難易度を大きく引き上げた要因の一つである。通常、ストレージエンジンの開発は一定のインタフェース内で完結することが期待されるが、本プロジェクトではその前提が成立せず、MySQL 本体の挙動を理解した上で、どの部分を変更し、どの部分を維持するかという判断が求められた。

さらに、このような変更は単独で完結するものではなく、可用性設計やフェイルオーバー構成にも影響を与えるため、コア部分とシステム全体の設計を同時に調整する必要があった。このような相互依存関係の中で整合性を保ちながら

開発を進めることが、本プロジェクトにおける大きな難しさであった。

次に可用性設計の領域では、中森氏が中心となり、レプリケーションおよび障害対応の構成が検討された。ここでは、データの同期方法や障害発生時の挙動をどのように設計するかが主要な課題となっている。特に、どの時点で障害と判定するか、どのノードに切り替えるか、切り替え後のデータ整合性をどのように保証するかといった点で、複数の方式が検討されている。

フェイルオーバーの設計においては、単純に切り替えを行うだけではなく、切り替えに伴う遅延や整合性の崩れを最小限に抑える必要がある。そのため、障害検知の閾値設定や、切り替え手順の設計、データ同期のタイミングといった複数の要素が密接に関係しており、調整には多くの試行錯誤が伴っている。

運用構成の領域では、李氏が中心となり、Orchestrator や ProxySQL を用いた構成が構築された。これらのツールを組み合わせることで、障害発生時に自動的に接続先を切り替える仕組みが実現されているが、実際にはツール間の連携や設定の整合性に多くの調整が必要であった。例えば、障害検知のタイミングと接続切り替えのタイミングが一致しない場合、一時的に不整合な状態が発生することがあり、その解消には設定の微調整が必要となった。

さらに、これらの可用性設計とコア部分の実装は独立しているわけではなく、相互に影響を与える。コアの処理モデルの変更がレプリケーションに影響を与えたり、フェイルオーバーの方式が性能に影響を与えたりするため、個別最適ではなく全体最適の観点での調整が求められた。

開発の後半では、これらの要素を統合するフェーズに移行している。この段階では、個別には動作していた機能同士を結合し、実際のシステムとして成立させるための調整が行われた。統合の過程では、想定していなかった問題が多数発生し、その都度原因を特定し、修正を行うという作業が繰り返されている。

最終的には、コアとなるデータ処理、可用性設計、運用構成が統合され、実際に動作するデータベースシステム「Kamo」として成立する段階に到達している。これは単なる機能の集合ではなく、複数の要素が相互に連携するシステムとして成立している点が重要である。

さらに、本プロジェクトでは、図 1 に示すとおり TPC-C ワークロードを用いた評価において、既存の MySQL と比較して大幅な性能向上が確認されている。Kamo を適用した構成では、従来構成では達成できなかった水準のスループットが実現されており、これは単なる最適化ではなく、内部構造に踏み込んだ設計変更が実際のワークロードにおいて有効に機能していることを示している。

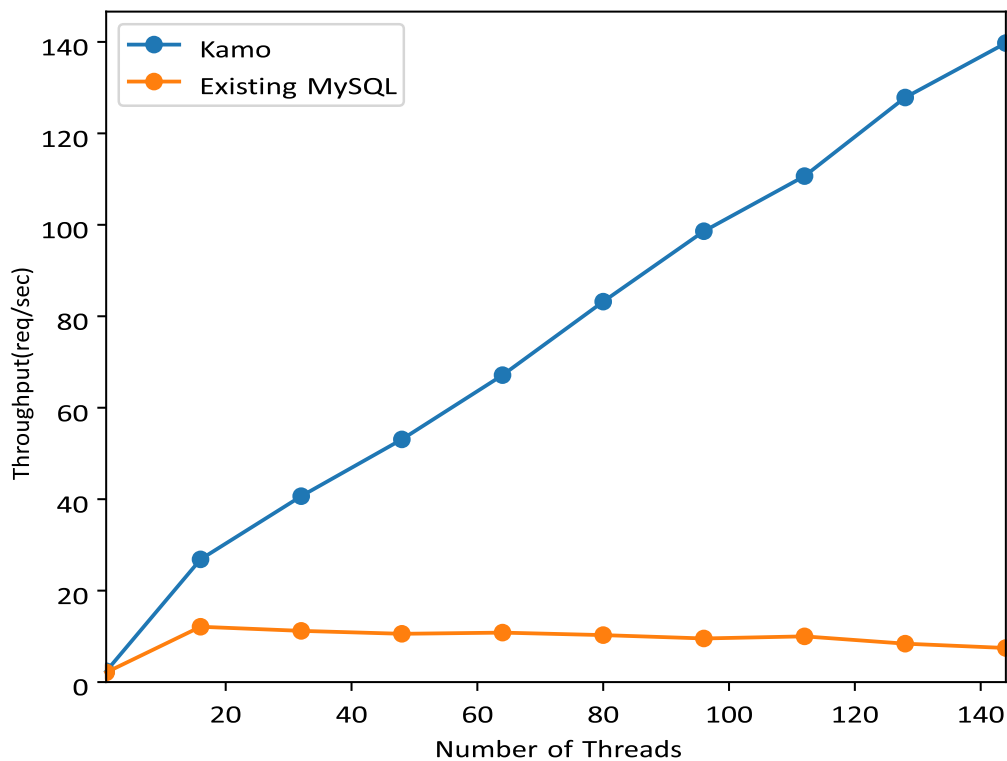


図 1 : ProxySQL 構成における TPC-C ワークロードでの Kamo と既存の MySQL の比較

この結果は、本プロジェクトが「動く」という段階にとどまらず、実用的な性能水準において成立していることを示すものであり、設計と実装の両面での到達を裏付ける重要な証拠である。

以上のように、本プロジェクトは既存の MySQL の内部構造に踏み込み、LinearDB の統合、可用性設計、運用構成の構築を並行して進め、それらを統合することで実用的なシステムとして成立させるに至っている。その過程では、設計上の前提の違いによる不整合や、フェイルオーバーの実装に伴う調整など、多くの試行錯誤が伴っており、これらを乗り越えて成立させた点が進捗の本質である。

## 10. プロジェクト評価

本プロジェクトの評価において重要なのは、新しいアルゴリズムや単体機能の開発ではなく、既存の複雑なデータベースシステムを内部から扱い、その構造を拡張・統合することで実用的なシステムとして成立させた点にある。

MySQL は長年にわたり改良が重ねられてきた成熟したシステムであり、その内部には多くの前提条件や最適化が組み込まれている。本プロジェクトでは、その中でも特に InnoDB に最適化された構造が強く影響しており、ストレージエンジンの抽象化レイヤが存在するにもかかわらず、実際には InnoDB 前提の処理が各所に存在しているという問題に直面した。

このため、単に新しいストレージエンジンを実装するだけでなく、MySQL

本体のコードに対しても修正を加えながら統合を進める必要があった。この点は、通常の拡張開発とは質的に異なる難しさを持っており、本プロジェクトの難易度を大きく引き上げている要因であった。

また、LinearDB の統合においても、既存のトランザクション処理やロック機構との整合性を保ちながら設計を行う必要があり、理論と実装の間を往復しながら調整が進められた。単に動作させるだけでなく、性能と整合性を同時に成立させる必要があるため、設計判断の難易度は高かった。

さらに、本プロジェクトは単一の技術領域に閉じたものではなく、コアとなるデータ処理、可用性設計、運用構成を統合することで初めて成立するシステムである。このため、各要素の最適化だけでなく、全体としての整合性を維持するための調整が求められており、この点も本プロジェクトの特徴である。

宮崎氏はコア部分の設計および実装を主導し、既存の MySQL 内部構造に踏み込んだ改修を行っている点で、プロジェクトの中核を担った。一方で、中森氏および李氏は、それぞれ可用性設計および運用構成において重要な役割を果たしており、システムとして成立させる上で不可欠な貢献を行った。

本プロジェクトは、構造そのものを新たに発明することで未踏性を示すタイプのプロジェクトではなく、「既存の複雑な構造を扱い切り、成立させることによって価値を生み出す」タイプのプロジェクトである。その意味で、アルゴリズム的な新規性とは異なる次元の難しさを持つものであり、この点を評価する必要がある。

以上より、本プロジェクトは、既存システムの制約に対して内部構造レベルでアプローチし、複数の要素を統合することで実用的なシステムとして成立させた点において、技術的難易度および実装力の両面から高く評価できる。

## 11. 今後の課題

本プロジェクトは、既存の MySQL の内部構造に踏み込み、拡張および統合によって実用的なシステムとして成立する段階に到達しているが、今後の発展に向けてはいくつかの重要な課題が残されている。

第一に、設計の独立性に関する課題である。本プロジェクトでは MySQL 本体、とりわけ InnoDB に最適化された構造に対して改修を加える形で統合を行っているが、このようなアプローチは既存システムへの依存度が高くなる傾向がある。今後、他の環境や構成への適用を考えた場合、どの程度まで汎用化できるか、あるいはどの部分を切り出して独立した設計とするかが課題となる。

第二に、スケーラビリティおよび性能検証の課題である。現状は一定規模の環境において動作が確認されているが、より大規模な環境や異なる負荷条件において同様の性能および可用性が維持されるかについては、さらなる検証が必要である。特に、負荷が増大した際のボトルネックや、障害発生時の挙動については、実運用に近い条件での評価が求められる。

第三に、可用性設計の成熟度に関する課題である。フェイルオーバーやレプリケーションの構成は実装されているが、障害検知の精度、切り替えのタイミング、整合性の保証といった点については、さらなる調整が必要である。特に、複数の要素が相互に影響する構造であるため、個別最適ではなく全体最適の観点から設計を洗練させる必要がある。

第四に、運用および保守性の課題である。MySQL本体への改修を伴う構成は、機能として成立している一方で、将来的なバージョンアップや環境変更に対する追従性、運用時のトラブルシューティングの容易さといった観点で課題が残る可能性がある。実運用を前提とする場合、性能や機能だけでなく、運用性および保守性を含めた設計が重要となる。

以上より、本プロジェクトは既存のデータベースシステムに対して内部構造レベルでアプローチし、統合されたシステムとして成立させるところまで到達しているが、今後は汎用性、スケーラビリティ、可用性設計の成熟度、運用性といった観点での発展が求められる段階にある。