MySQL 5.5.16 リリースノート

外部認証

- 現在、MySQL の商用版には、MySQL サーバで外部認証メソッドを使用して MySQL ユーザを認証できる、2 つのプラグインが搭載されている。
  - PAM（Pluggable Authentication Modules:プラガブル認証モジュール）により、システムで標準インタフェースを使用し各種認証メソッドにアクセスできる。PAM 認証プラグインによって、MySQL サーバでは PAM の使用で MySQL ユーザを認証することが可能になる。

    PAM プラグインでは、MySQL サーバによりプラグインに渡される情報（ユーザ名、ホスト名、パスワード、および認証文字列など）、さらに PAM ルックアップ（Unix パスワード、または LDAP ディレクトリなど）に対して利用可能なものは何でも使用される。プラグインは PAM に対するユーザ証明書をチェックし、成功か失敗を返す。

    PAM 認証プラグインは、Linux および Mac OS X で検証済みである。

    > 注意事項
    >
    > PAM プラグインは、クライアントサイドのプラグインで機能するが、クライアントサイドのプラグインは、単純にクリアテキストのパスワードをサーバに送信し、それが PAM に渡される。そのため、設定によってはセキュリティ上の問題になる場合があるが、サーバサイドの PAM ライブラリの使用は必要である。パスワードが盗まれる可能性がある場合、問題を回避するために、クライアントは SSL を使用して MySQL サーバに接続する必要がある。詳細はセクション 5.5.6.4「The Clear-Text Client-Side Authentication Plugin」を参照。
  - MySQL の Windows 用ディストリビューションには、MySQL サーバで Windows のネイティブなサービスを使用してクライアント接続を認証できる認証プラグインが搭載されている。Windows にログインしたユーザは追

加のパスワードを指定しなくても、ユーザ環境の情報を基に、MySQL クライアントプログラムからサーバに接続可能できる。

クライアントとサーバは、認証ハンドシェイクでデータパケットをやり取りする。このやり取りの結果、サーバは、Windows OS のクライアントの識別を示すセキュリティコンテキストオブジェクトを作成する。この識別には、クライアントアカウントの名前が含まれている。Windows 認証プラグインでは、クライアントの識別を使用して、それが特定のアカウントやグループのメンバであるかどうかをチェックする。ネゴシエーションはデフォルトで認証に Kerberos を使用し、Kerberos が使えない場合は NTLM を使用する。

Windows の認証プラグインは、Windows 2000 Professional で稼動するものとする。

これらの認証プラグインにより、MySQL サーバでは、MySQL 権限テーブル以外に定義されたユーザからの接続を受け入れることができる。これらのプラグインは、MySQL プロキシ-ユーザ機能もサポートしている。各プラグインは、ログインユーザとは異なるユーザ名を MySQL に返すことがあるが、これはプラグインが返す MySQL ユーザには外部認証のユーザに必要な権限が定義されていることを意味する。たとえば、joe という名前の外部ユーザが接続して developer という名前の MySQL ユーザの権限を持つ場合がある。

サーバサイド PAM および Windows の認証プラグインは商用版にのみ搭載されており、MySQL コミュニティ版には搭載されていない。これらプラグインが通信するクライアントサイドのプラグインは、コミュニティ版も含むすべての版に搭載されている。そのため、どのリリースのクライアントでも、サーバサイドのプラグインがロードされたサーバへの接続が許可されている。

これらのプラグインについての詳細は、セクション 5.5.6.2「The PAM Authentication Plugin」、およびセクション 5.5.6.3「The Windows Native Authentication Plugin」を参照。MySQL のプラガブル認証の一般的な情報については、セクション 5.5.6「Pluggable Authentication」を参照。プロキシユーザの詳細は、セクション 5.5.7「Proxy Users」を参照。

スレッドプールプラグインに関する注意事項

- MySQL サーバのデフォルトのスレッド処理モデルでは、クライアント接続ごとに 1 つのスレッドを使用してステートメントを実行する。サーバへ接続してステートメントを実行するクライアントが増えるほど、全体的なパフォーマンスは低下する。MySQL の商用版には、現在それに代わるスレッド処理モデルを導入したスレッドプールプラグインが搭載されている。この処理モデルはオーバヘッドを削減し、パフォーマンスが向上するように設計されている。プラグインに実装されているスレッドプールは、大量のクライアント接続に対するステートメント実行スレッドを効率的に管理して、サーバのパフォーマンスを向上させる。

  スレッドプールは、接続ごとに 1 つのスレッドを採用するモデルにおける、以下のような問題に対処している。

  - 多すぎるスレッドスタックは、並行実行の作業負荷が高くなり CPU キャッシュをほとんど使用不可能な状態にする。スレッドプールでは、CPU キャッシュのフットプリントを最小限に抑えるために、スレッドスタックの再使用を推進している。
  - 並行実行するスレッドが多すぎると、コンテキスト切り替えのオーバヘッドが増大する。これはオペレーティングシステムスケジューラでも問題になる。スレッドプールは、MySQL サーバ内の並列性を保てるようにアクティブなスレッドの数を制御する。このときに適用されるのは、サーバによる処理が可能で MySQL が実行中のサーバホストに適したレベルである。
  - 並列実行するトランザクションの数が多すぎると、リソースの競合が増大する。InnoDB では、主要なミューテックスを保持するために費やす時間が増加する。スレッドプールは、並列実行が増えすぎないようにトランザクション開始の時期を制御する。

  Windows の場合、スレッドプールプラグインを使用するために Windows Vista 以降が必要。Linux の場合、kernel 2.6.9 以降が必要。

  詳細はセクション 7.11.6「The Thread Pool Plugin」を参照。

機能の追加または変更

- ***重要な変更: レプリケーション***: RESET SLAVE ステートメントが ALL キーワードで拡張された。RESET SLAVE ALL は、master.info、relay-log.info およびすべての

リレーログファイルの削除に加えて、それ以外に RESET SLAVE の実行後にメモリ内に保存されるすべての接続情報も削除する（Bug #11809016）。

- スレッドプールプラグインはサーバの起動時にロードし、実行時にはロードまたはアンロードしないようにする必要がある。現在は、INSTALL PLUGIN または UNINSTALL PLUGIN ステートメントで、スレッドプールプラグインをロードまたはアンロードしようとするとエラーが発生する。
- 一部のプラグインは、サーバの起動時にロードし、実行時にはロードまたはアンロードしないような状況で動作する。プラグイン API は、現在このようにマーキングしたプラグインをサポートしている。st_mysql_plugin 構造には flags メンバが組み込まれ、該当するフラグの OR に設定できる。PLUGIN_OPT_NO_INSTALL フラグは、INSTALL PLUGIN ステートメントを実行時に指定しても、このプラグインをロードできないことを示す。このフラグは、--plugin-load オプションでサーバ起動時にロードする必要があるプラグインに適している。PLUGIN_OPT_NO_UNINSTALL フラグは、実行時に UNINSTALL PLUGIN ステートメントを指定しても、このプラグインをアンロードできないことを示す。

  新規メンバによりインタフェースが変更されたので、プラグインのインタフェースバージョン MYSQL_PLUGIN_INTERFACE_VERSION は、0x0102 から 0x0103 にアップした。新規メンバにアクセスする必要があるプラグインは、バージョン 0x0103 以上を使用するために再コンパイルする必要がある。

- 新しいユーティリティ *mysql_plugin* により、MySQL 管理者は、どのプラグインを MySQL サーバにロードするかを管理できる。別の手段も搭載されており、サーバ起動時に手動で--plugin-load オプションを指定したり、実行時に INSTALL PLUGIN および UNINSTALL PLUGIN ステートメントを使用することもできる。詳細はセクション 4.4.4「mysql_plugin — Configure MySQL Server Plugins」を参照。

修正されたバグ

- **InnoDB ストレージエンジン**: この修正により、InnoDB テーブルの VARCHAR(*N*) カラムの操作パフォーマンスが向上する。ここで *N* は大きな値として宣言されるが、テーブルの実際の文字列値の長さは短い（Bug #12835650）。
- **InnoDB ストレージエンジン**: 使用されていない関数は、論理を明確にするため、小さなトランザクションに関連する内部 InnoDB コードから削除された（Bug #12626794、Bug #61240）。

- **InnoDB ストレージエンジン**: InnoDB プラグインから削除された「random read-ahead」機能を再び使用できる。この機能は特定のワークロードにのみ有用なため、デフォルトではオフになっている。オンにするには、innodb_random_read_ahead 構成オプションを有効にする。この機能は、場合によりパフォーマンスを向上させたり低下させたりするので、この設定を使用する前に、設定を有効および無効の両方でベンチマークを実行する（Bug #12356373）。
- **非互換の修正**：日付関連のアサーションの取り扱いが修正された。しかし、この修正の結果として、いくつかの機能は DATE() 関数に引数が渡されるときに、チェックがより厳格になり、日付部分が 0 で埋められていないものを拒否するようになった。影響を受ける関数は、CONVERT_TZ(), DATE_ADD(), DATE_SUB(), DAYOFYEAR(), LAST_DAY(), TIMESTAMPDIFF(), TO_DAYS(), TO_SECONDS(), WEEK(), WEEKDAY(), WEEKOFYEAR(), YEARWEEK(). である。これが General Availability-status シリーズ（MySQL 5.1 と 5.5）において日付の取り扱いのふるまいを変えるので、この変更は 5.1.62 と 5.5.21 で前の状態に戻された。この修正は、MySQL 5.6 系で保持される。
- **InnoDB ストレージエンジン**:INFORMATION_SCHEMA.TABLES テーブルの中の DATE_LENGTH の列が、圧縮された InnoDB テーブルのディスク上のテーブルスペースのサイズを正確にレポートするようになった。(Bug #12770537)
- **InnoDB ストレージエンジン**:innodb_file_per_table=1 と innodb_file_format=Barracuda の設定により、ページサイズの半分より大きい値の列を挿入し、そしてその列がセカンダリーインデックスを含み、その列の値が更新されたときにクラッシュを引き起こすことができた。(Bug #12637786)
- **InnoDB ストレージエンジン**:ロジックをクリアにするために、ミニトランザクションに関する InnoDB 内部のコードから使わない関数を削除した。(Bug #12626794, Bug #61240)
- **レプリケーション**: 破損したテーブルマップイベントの処理を行うと、サーバがクラッシュする可能性があった。これは、Bug#56226 で発生する場合と同じで、イベントによって別々のテーブルが同じ ID にマップされる場合に、特に発生しやすかった。

  現在はサーバによりテーブルマップイベントが適用される前に、そのテーブルが異なる設定でマップされていたかどうかがチェックされる。もし該当する場合には、エラーが発生し、スレーブ SQL スレッドは停止する。テーブルが同じ設定でマップされていた場合、あるいはフィルタリングルールに無視されるようにテーブルが設定されている場合、動作は変更しない。イベントはスキップされ、ID はチェックされない（Bug #44360、Bug #11753004）。

  Bug #11763509 も参照。

- .frm または.TRG ファイルのみを開いて処理され、テーブルを数多くスキャンする必要があった INFORMATION_SCHEMA クエリに対して、メタデータロックサブシステムがオーバヘッドを増やし過ぎていた。たとえば、SELECT COUNT(*) FROM INFORMATION_SCHEMA.TRIGGERS に影響があった（Bug #12828477）。
- Mac OS X 10.7（Lion)で以下のような警告が表示され、コンパイルが失敗した。Implicit declaration of function 'pthread_init'（Bug #12779790）。
- 無効またはコンパイルされていないプロファイリングを使用すると、set_thd_proc_info()がファイル名の長さに関して不必要なチェックを実行した（Bug #12756017）。
- Bug #11792200 で追加された DBUG_ASSERT が、表明の発生に対して過剰だった（Bug #12537160）。
- CHECK TABLE および REPAIR TABLE は、基礎となるテーブルが欠けている MERGE テーブル、または不正なストレージエンジンを使用して問題のルックアップに失敗した。最初の基礎となるテーブルに関してのみ問題が報告されていた（Bug #11754210）。
- （5 DIV 2）と（5.0 DIV 2）が異なる結果（2 と 3）を生成していた。これは（5.0 DIV 2）の結果が、切り捨てされずに整数に変換されていたためである。このバグは、MySQL 5.0 および 5.1 の動作とは異なる。現在はいずれの式も 2 を生成する（Bug #61676、Bug #12711164）。
- lower_case_table_names の値が 1 または 2 で、データベース名に大文字小文字が混在している場合、そのデータベース名が含まれた完全修飾名でストアドファンクションを呼び出すと失敗した（Bug #60347、Bug #11840395）。
- argc = 0 の場合に、埋め込みサーバがクラッシュした（Bug #57931、Bug #12561297）。
- WHERE 句で決定的なストアドファンクションを使用して SELECT DISTINCT を指定すると、誤った結果になる可能性があった（Bug #59736、Bug #11766594）。
- RPM パッケージを使用してアップグレードを行うと、test データベースが再作成されていた。これは、このデータベースがすでに DBA により削除されていた場合は適切ではない（Bug #45415、Bug #11753896）。
- mysql_affected_rows() C API 関数が、重複キー値が存在する INSERT ... ON DUPLICATE KEY UPDATE ステートメントに対して（2 ではなく）3 を返していた（Bug #46675、Bug #11754979）。
- ENGINE オプションを指定しない CREATE TABLE は、実行時ではなく解析時にデフォルトエンジンを判別していた。そのため、このステートメントがストアドプログラム内で実行され、その間にデフォルトエンジンが変更されていた場合は不正な結果を導いていた（Bug #50614、Bug #11758414）。

※本翻訳は、理解のための便宜的な訳文として、オラクルが著作権等を保有する英語原文を NRI の責任において翻訳したものであり、変更情報の正本は英語文です。また、翻訳に誤訳等があったとしても、オラクルには一切の責任はありません。