# OSS DBMS の監査機能について

2012 年 3 月 27 日

第 1.0.1 版

SCSK(株)

OSS 基盤技術センター

OSS 第一技術課

# 改定履歴

| 改定日 | 版番号 | 内容 | 備考 |
|---|---|---|---|
| 2011/11/24 | 1.0.0 | 初版作成　（担当：田中） | |
| 2012/03/27 | 1.0.1 | 「4.4 MySQL の監査機能概要」に Audit Plugin に関する記述を追加 | （担当：田中） |
| | | | |

# 目次

**OSS DBMS の監査機能について ..... 1**

**1. はじめに ..... 1**

**1.1. 本書の位置づけ ..... 1**

**1.2. 前提知識 ..... 1**

**2. 背景 ..... 3**

**3. 調査対象 ..... 4**

**4. 各 DBMS の監査機能概要 ..... 5**

**4.1. Oracle Database の監査機能概要 ..... 5**

**4.2. PPAS の監査機能概要 ..... 6**

**4.3. PostgreSQL の監査機能概要 ..... 8**

**4.4. MySQL の監査機能概要 ..... 12**

**5. 具体的に取得できる情報 ..... 13**

**5.1. Oracle Database の場合 ..... 13**

**5.2. PPAS の場合 ..... 16**

**5.3. PostgreSQL の場合 ..... 16**

**5.4. MySQL の場合 ..... 17**

**6. 監査ログの保全 ..... 18**

**6.1. Oracle Database の場合 ..... 18**

**6.2. PPAS の場合 ..... 18**

**6.3. PostgreSQL の場合 ..... 19**

**6.4. MySQL の場合 ..... 19**

**7. 監査ログの運用 ..... 20**

**7.1. Oracle Database の場合 ..... 20**

**7.2. PPAS の場合 ..... 20**

**7.3. PostgreSQL の場合 ..... 20**

**7.4. MySQL の場合 ..... 21**

**8. まとめ ..... 22**

**8.1. 監査のしやすさ ..... 22**

**8.2. ログの容量 ..... 22**

**8.3. ログのセキュリティ対策 ..... 23**

**9. 今後の課題 ..... 24**

**10. 付録 ..... 25**

**10.1. 参考資料 ..... 25**

**10.2. 関連サイト ..... 25**

# 1. はじめに

## 1.1. 本書の位置づけ

本書は OSS 基盤技術センターで OSS の DBMS の監査機能を調査し、その結果をまとめたものです。

また、記述している項目については動作検証をおこなっておりますが、監査機能を利用した場合の処理パフォーマンスへの影響に関しては、検証できておりません。検証を行い順次情報を提供する予定です。

## 1.2. 前提知識

DBMS、Linux などについての知識が前提になります。

# 2. 背景

システムを運用していく上で、正しく運用されているか、セキュリティが確保されているかといったことが、金融商品取引法の内部統制報告(通称 J－SOX 法)への対応など、コンプライアンスに関する意識が高まってきている中で重要視されています。
特に中核となるデータベースに関しては、それらのコンプライアンス対応のため、「いつ、誰が、どこから、どのような操作を行ったか」を記録して管理できることが求められることとなります。(監査ログに対する基準は、「情報セキュリティ管理基準 (平成 20 年改正版) 平成 20 年経済産業省告示第 246 号」P.56 の「6.10 監視」などをご参照ください。)

それを実現するためには、データベースへのアクセスに関して、監査を行うことができるような情報を記録し保存する仕組みを用意しておくことが必要です。

そこで、OSS の DBMS を利用した場合に、DBMS の機能としてどこまで監査のために利用できる情報を取得して管理することができるかを確認し、Oracle Database を利用して運用しているシステムと同様のことが実現できるかを確認しました。

# 3. 調査対象

今回の調査対象としては､OSS である PostgreSQL と MySQL､そして PostgreSQL をベースに Oracle Database との互換機能やパフォーマンス改善の機能を盛り込んだ Postgres Plus Advanced Server (以下、PPAS)を取り上げることとしました。

これらは、日本国内でもいくつかの会社がサポートサービスを提供している OSS のプロダクトです。PPAS に関しては、商用パッケージではありますが、Oracle Databse との互換性が高いといわれているため、Oracle Database からの移行を考慮した場合に、移行工数を削減できることが期待できるプロダクトであるため取り上げることとしました。

また、比較対象として Oracle Database の機能も確認することで、OSS や OSS をベースにしたプロダクトで取得できるものとの比較を行い、Oracle Database から移行する場合にどの程度まで実現できるかを確認しています。

具体的には、それぞれ以下のバージョンを利用して確認しました。

**表 3-1 調査対象の DBMS プロダクト**

| プロダクト名 | バージョン |
|---|---|
| Oracle Database 11g R2 | 11.2.0.1 |
| Postgres Plus Advanced Server | 9.0.4 |
| PostgreSQL | 9.0.4 |
| MySQL | 5.5.17 |

# 4. 各 DBMS の監査機能概要

## 4.1. Oracle Database の監査機能概要

比較対象として、まずは、Oracle Database がどこまでの監査機能を持っているかを確認しておきます。

Oracle Database では、標準で以下の監査の機能が用意されています。

**表 4-1 Oracle Database の監査機能**

| 名称 | | 概要 |
|---|---|---|
| 必須監査 | | インスタンスの起動と停止や管理者権限によるインスタンスへの接続に関するログの出力を行います。<br>リスナーに対する接続と接続エラーに関しても、同様にログが出力されます。 |
| DBA 監査 | | SYS/SYSDBA/SYSOPER 権限で行われた全ての操作をログに出力します。 |
| 標準監査 | 文監査 | 実行する SQL 文の種別によって監査します。 |
| | 権限監査 | 権限を利用した際の特定の権限による操作を監査します。 |
| | オブジェクト監査 | 特定のオブジェクトへの操作を監査します。 |
| ファイングレイン監査 | | ログを出力する条件として、監査する対象のカラムの値や範囲で指定するなど詳細な条件を付与することで、その条件に合致するものだけの監査ログを取得します。 |

これらの監査機能のうち最も特徴的なのは、ファイングレイン監査です。これを利用することで、必ず監査しなければならないような対象のレコードやカラムに対する操作だけピックアップして監査ログを取得することができます。

出力先に関しては、AUDIT_TRAIL 初期化パラメータで指定することができ、「DB」「DB, EXTENDED」「OS」「XML」「XML, EXTENDED」「NONE」といった指定が可能で、11gR2 ではデフォルトで「DB」が設定されています。

ただし、このファイングレイン監査を利用するためには、Oracle Database の Enterprise Edition であることが必要です。

## 4.2. PPAS の監査機能概要

PPAS には、独自の監査の機能としてログファイルに監査ログを出力する機能が用意されています。

設定によって、データベースやテーブルに対する変更や、テーブルへの更新、テーブルの参照など、いくつかの段階での指定ができ、そのログを指定したディレクトリにファイルとして出力することができます。

この機能に対して設定できる項目は、以下の項目です。

**表 4-2 PPAS の監査機能で設定できる項目**

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| edb_audit | none | 監査ログを取得するかどうか、取得する場合に XML で出力するか CSV で出力するかを選択することができます。<br>「none」は取得しない、「xml」は XML 形式で出力する、「csv」は CSV 形式で出力することを指定します。 |
| edb_audit_directory | edb_audit | 監査ログを出力する出力先のディレクトリを指定します。<br>デフォルトでは、PGDATA ディレクトリの下に「edb_audit」というサブディレクトリを作成して、そこにファイルを出力します。 |
| edb_audit_filename | audit-%Y-%m-%d_%H%M%S | 出力するファイル名を指定します。<br>ファイル名に strftime()で利用できるフォーマットを利用することができます。<br>デフォルトの設定では、「audit-2011-10-01_162430」となるようなフォーマットが指定されています。そして、実際のファイル名は、CSV フォーマットなら拡張子として「.csv」、XML フォーマットなら「.xml」が付与されます。 |

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| edb_audit_rotation_day | every | 曜日でのファイルのローテーションを指定します。<br>「none」はローテーションしない、「every」は毎日、「sun」「mon」…「sat」など曜日を指定してローテーションすることを指定できます。 |
| edb_audit_rotation_size | 0 | ファイルサイズでのファイルのローテーションを MB 単位で指定します。<br>デフォルトは 0 なので、サイズによるファイルローテーションは行いません。 |
| edb_audit_rotation_seconds | 0 | 秒数でのファイルのローテーションを指定します。<br>デフォルトは 0 なので、秒数によるファイルのローテーションは行いません。 |
| edb_audit_connect | failed | 接続時のログを出力するかを指定します。<br>「none」「failed」「all」 |
| edb_audit_disconnect | none | 切断時のログを出力するかを指定します。<br>「none」「all」 |
| edb_audit_statement | ddl,, error | 実行された SQL 文の種別によって出力するかを指定します。<br>「none」「dml」「ddl」「select」「error」「all」 |

これらの設定は、postgresql.conf に設定します。postgresql.conf 内に記述するので、サーバ起動時にのみ読み込ませて設定することができます。サーバ起動後に動的に変更することはできません。

ただし、設定項目からもわかるとおり、監査用のログを出力する監査対象のデータベースやテーブルなどを細かく指定する設定が用意されていないため、取得する場合は DBMS 全体に対する設定を行うことになります。

## 4.3. PostgreSQL の監査機能概要

PostgreSQL には、発行された SQL 文をログファイルとして記録するための機能があります。この機能は、PostgreSQL をベースにしている PPAS でも利用することができます。

この機能に対して設定できる項目は、以下の項目になります。

**表 4-3 PostgreSQL のログファイル出力で設定できる項目**

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| log_destination | stderr | ログの出力方法や形式を指定します。<br>「syslog」を指定すると、syslog を利用してログを送出します。「stderr」を指定すると標準エラー出力に出力されます。「csvlog」を指定すると CSV 形式でログを出力するようになります。 |
| logging_collector | on | ログを取得するかどうかを On/Off で指定します。<br>「on」を指定するとログを出力するようになります。 |
| log_directory | pg_log | ログの出力先ディレクトリを指定します。<br>データディレクトリからの相対パスか絶対パスで指定できます。 |
| log_filename | postgresql-%a.log | 出力するファイル名を指定します。<br>ファイル名には、strftime()のフォーマットを利用できます。「%a」は、曜日の省略名に置換されます。 |
| log_truncate_on_rotation | on | ローテーション時に上書きしてしまうかどうかを指定します。<br>例えば、曜日ごとにファイルに出力して 1 週間単位でログをローテーションするような場合に利用します。 |

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| log_rotation_age | 1d | 期間(分単位)によるログのローテーションを指定します。<br>1d と指定すると 1 日毎のローテーションとなります。 |
| log_rotation_size | 0 | ログファイルのサイズ(KB 単位)によるローテーションを指定します。<br>0 を指定した場合は、ファイルサイズでのローテーションを行いません。 |
| client_min_messages | notice | クライアント側に返却するメッセージのレベルを指定します。<br>「debug5」「debug4」「debug3」「debug2」「debug1」「log」「notice」「warning」「error」などが指定できます。 |
| log_min_messages | warning | ログに出力するメッセージのレベルを指定します。<br>「debug5」「debug4」「debug3」「debug2」「debug1」「info」「notice」「warning」「error」「log」「fatal」「panic」などが指定できます。 |
| log_min_error_statement | error | エラーが発生した時のエラーの原因となった SQL をどのレベルのエラーまで出力するかを指定します。<br>「debug5」「debug4」「debug3」「debug2」「debug1」「info」「notice」「warning」「error」「log」「fatal」「panic」などが指定できます。 |
| log_autovacuum_min_duration | -1 | autovacuum に多くの時間がかかっていないか長い処理時間になっていないかを出力する閾値を秒数で指定します。<br>-1 を指定すると出力しません。 |
| log_min_duration_statement | -1 | 処理の遅いクエリを出力するための閾値をミリ秒単位で指定します。<br>-1 を指定すると出力しません。 |

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| log_checkpoints | off | チェックポイントとリスタート時にログを出力するかを指定します。<br>「on」「off」 |
| log_connections | off | 接続時のログを出力するかを指定します。<br>「on」「off」 |
| log_disconnections | off | 切断時のログを出力するかを指定します。<br>「on」「off」 |
| log_duration | off | 実行された文に関して経過時間をログ出力するか指定します。<br>「on」「off」 |
| log_error_verbosity | default | ログに出力するメッセージの詳細度を指定します。<br>「terse」「default」「verbose」 |
| log_hostname | off | ログに出力する際のホスト名を名前解決して出力するかを指定します。<br>「on」「off」 |
| log_line_prefix | ‘’ | ログをファイルに出力する際の各行の最初に出力する値をフォーマットで指定します。<br>デフォルトは空文字です。<br>空文字のままだと監査で必要となる情報が取得できない場合があるので、必要となる情報を出力するように指定する必要があります。 |
| log_lock_waits | off | deadlock_timeout よりもロック獲得に時間がかかる場合のログを出力するか指定します。<br>「on」「off」 |

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| log_statement | none | ログに出力する対象の SQL 文を指定します。<br>「none」を指定すると出力しません。「ddl」を指定すると CREATE、ALTER、DROP などの DDL を出力します。「mod」を指定すると DDL の出力に INSERT、UPDATE、DELETE なども出力します。「all」では全ての SQL 文を出力します。ただし、明らかに構文的に誤っている SQL 文が実行された場合はログに出力されません。こういったエラーも出力させたい場合は、log_min_error_statement を ERROR よりもより詳細な情報が出力される値に変更してください。 |
| log_temp_files | -1 | 作成される一時ファイルに対するログの出力有無とサイズを指定します。<br>「0」を指定すると全ての一時ファイルの情報をログに出力し、キロバイト単位で数値を指定するとそのサイズを超える一時ファイルについてログを出力します。「-1」を指定することでこの一時ファイルに対するログを出力しないようになります。 |

これらの設定は、postgresql.conf に行います。

また、postgresql.conf で設定する以外にも、コマンドによって、特定のユーザや特定のデータベースのみを指定してログを出力するよう指定することができます。

例えば、PostgreSQL 管理用ユーザ「postgres」はログを出力せず、アプリケーション用の DBA ユーザである「testdba」のオペレーションを全て残し、アプリケーション用の通常ユーザはデータの変更に関するログを残すというような場合は、以下のようなコマンドを実行します。

```
ALTER ROLE postgres SET log_statement TO 'none';
ALTER ROLE testdba SET log_statement TO 'all';
ALTER ROLE testuser SET log_statement TO 'mod';
```

さらに、引数に「IN database データベース名」も以下のように追加すると、特定のデータベースのみに関するログを取得するように設定できます。

```
ALTER ROLE testuser IN database testdb set log_statement to 'all';
```

また、log_line_prefix に監査に必要となると思われる項目を追加することで、例えば、タイムスタンプ(ミリ秒付き)、データベース名、トランザクション ID、リモートホスト名、セッション識別子などの情報を付与した形でファイルに出力することができます。ただし、log_destination に csvlog を指定している場合は、このプレフィックスの部分が固定化されていて、log_line_prefix を変更しても反映されませんのでご注意ください。

## 4.4. MySQL の監査機能概要

MySQL の場合は、5.1.6 よりも前のバージョンの場合、オペレーションのログをファイルに出力することが可能です。

5.1.6 以降のバージョンであれば、さらにそのログを DB 上に保存することも可能になっています。

さらに、MySQL 5.5 からは、Audit Plugin が用意され監査機能を拡張できますが、本資料では、標準実装の監査機能についてまとめます。

- ✓ (追記 2012/11/07) 2012 年 10 月以降、MySQL Enterprise Edition には MySQL Enterprise Audit という機能が用意されているそうです。

この機能に対して設定できる項目は、以下の通りです。

**表 4-4 MySQL のログファイル出力で設定できる項目**

| 項目名 | デフォルト値 | 概要 |
|---|---|---|
| log-output | TABLE | ログの出力形式を指定します。<br>「TABLE」を設定すると DB 上にログを出力します。「FILE」を指定するとファイルとして出力します。「NONE」を指定するとログを出力しません。<br>「TABLE,FILE」と指定して、DB とファイルの両方に出力させることもできます。 |
| general-log | | ログ出力の有効無効を指定します。<br>「ON」「OFF」 |

これらの設定は、/etc/my.cnf の[mysqld]セクションに記述して指定します。

# 5. 具体的に取得できる情報

## 5.1. Oracle Database の場合

Oracle Database 11gR2 の場合、以下のような項目をもった情報がデフォルトで DB 上に監査情報として保存されています。

**表 5-1 Oracle Database で取得可能な監査情報**

| 列 | データ型 | NULL | 説明 |
|---|---|---|---|
| OS_USERNAME | VARCHAR2(255) | | 操作が監査対象となったユーザの OS 上のユーザ名 |
| USERNAME | VARCHAR2(30) | | 操作が監査対象となったユーザの名前 |
| USERHOST | VARCHAR2(128) | | ユーザが Oracle インスタンスからデータベースにアクセスしている場合の Oracle インスタンスの数値 ID |
| TERMINAL | VARCHAR2(255) | | ユーザの端末の識別子 |
| TIMESTAMP | DATE | NOT NULL | 監査証跡エントリの作成または CONNECT 文のログイン時刻のタイムスタンプ |
| OWNER | VARCHAR2(30) | | 操作の影響を受けたオブジェクトの作成者 |
| OBJ_NAME | VARCHAR2(128) | | 操作の影響を受けたオブジェクトの名前 |
| ACTION_NAME | VARCHAR2(28) | | DBA_AUDIT_TRAIL の ACTION 列の数値コードに対応する操作タイプの名前 |
| NEW_OWNER | VARCHAR2(30) | | NEW_NAME 列に指定されたオブジェクトの所有者 |

| 列 | データ型 | NULL | 説明 |
|---|---|---|---|
| NEW_NAME | VARCHAR2(128) | | RENAME 後のオブジェクトの新規名、または基礎となっているオブジェクトの名前 |
| SES_ACTIONS | VARCHAR2(19) | | セッションのサマリー（16 文字で構成される文字列で、ALTER 、AUDIT 、COMMENT 、DELETE 、GRANT 、INDEX 、INSERT 、LOCK 、RENAME 、SELECT 、UPDATE 、REFERENCES 、EXECUTE の順に各操作の状態を 1 文字で表す。情報がない場合は-、成功の場合は S 、失敗の場合は F 、両方の場合は B 。） |
| COMMENT_TEXT | VARCHAR2(4000) | | 監査証跡についてのテキスト・コメント |
| SESSIONID | NUMBER | NOT NULL | 各 Oracle セッションの数値 ID |
| ENTRYID | NUMBER | NOT NULL | セッションの各監査証跡エントリの数値 ID |
| STATEMENTID | NUMBER | NOT NULL | 文の実行ごとの数値 ID |
| RETURNCODE | NUMBER | NOT NULL | 操作によって生成された Oracle エラーコード。有効な値の例は次のとおり。・0: 操作は成功・2004: セキュリティ違反 |
| PRIV_USED | VARCHAR2(40) | | 操作の実行に使用されたシステム権限 |
| CLIENT_ID | VARCHAR2(64) | | 各 Oracle セッションでのクライアント識別子 |
| ECONTEXT_ID | VARCHAR2(64) | | アプリケーション実行コンテキスト識別子 |
| SESSION_CPU | NUMBER | | 各 Oracle セッションで使用された CPU タイム |

| 列 | データ型 | NULL | 説明 |
|---|---|---|---|
| EXTENDED_TIMESTAMP | TIMESTAMP(6) WITH TIME ZONE | | UTC タイムゾーンでの監査証跡エントリで作成されたタイムスタンプ |
| PROXY_SESSIONID | NUMBER | | プロキシセッションシリアル番号 |
| GLOBAL_UID | VARCHAR2(32) | | ユーザのグローバルユーザ識別子 |
| INSTANCE_NUMBER | NUMBER | | INSTANCE_NUMBER 初期化パラメータで指定されたインスタンス番号 |
| OS_PROCESS | VARCHAR2(16) | | Oracle プロセスの OS のプロセス識別子 |
| TRANSACTIONID | RAW(8) | | オブジェクトがアクセスまたは変更されたトランザクションのトランザクション識別子 |
| SCN | NUMBER | | 問合せのシステム変更番号 |
| SQL_BIND | NVARCHAR2(2000) | | 問合せのバインド変数データ |
| SQL_TEXT | NVARCHAR2(2000) | | 問合せの SQL テキスト |
| OBJ_EDITION_NAME | VARCHAR2(30) | | 監査対象オブジェクトを含んでいるエディションの名前 |

デフォルトでは、DB 上に保存されていますので、情報を取得するために、DBA_AUDIT_OBJECT や USER_AUDIT_OBJECT などのビューを利用してアクセスします。

DB からの取得ですので、取り出す際に、条件を付与して一部の情報のみを取得して参照することもできます。

## 5.2. PPAS の場合

PPAS 独自の監査機能を利用した場合、監査情報として xml もしくは CSV 形式でファイルに出力されます。

項目としては、タイムスタンプ(ミリ秒付き)、ユーザ名、データベース名、プロセス ID、リモートホスト名、セッション ID、トランザクション ID、コマンドの種類、実行された SQL 文やコマンド、アプリケーション名が出力されます。

## 5.3. PostgreSQL の場合

PostgreSQL でログを出力するために log_destination に stderr を指定して、log_line_prefix をデフォルトの空文字のままにしていた場合は、実行した SQL 文やコマンドのみが監査情報としてファイルに出力されます。

これだけの情報では、監査するための情報としては不足なため、log_line_prefix に出力する項目を指定することで、ログに出力できる情報項目を追加・変更することができます。また、log_destination に csvlog を指定した場合は、デフォルトで、タイムスタンプ(ミリ秒付き)、ユーザ名、データベース名、プロセス ID、リモートホスト名、セッション ID、トランザクション ID、コマンドの種類、実行された SQL 文やコマンド、アプリケーション名が出力されるようになっていて、log_line_prefix の値は無視されます。

log_line_prefix に指定できる項目は、以下の通りです。

**表 5-2 log_line_prefix に指定可能なエスケープ文字列**

| エスケープ文字列 | 概要 | セッションプロセスのみ |
|---|---|---|
| %a | アプリケーション名 | ○ |
| %u | ユーザ名 | ○ |
| %d | データベース名 | ○ |
| %r | リモートホスト名または IP アドレス、ポート番号 | ○ |
| %h | リモートホスト名または IP アドレス | ○ |
| %p | プロセス ID | × |
| %t | タイムスタンプ | × |
| %m | タイムスタンプ(ミリ秒付き) | × |
| %i | コマンドの種類 | ○ |
| %e | SQLSTATE エラーコード | × |

| エスケープ文字列 | 概要 | セッションプロセスのみ |
|---|---|---|
| %c | セッション ID | × |
| %l | 各セッションまたは各プロセスのログの行番号 | × |
| %s | プロセスの開始タイムスタンプ | × |
| %v | 仮想トランザクション ID | × |
| %x | トランザクション ID | × |
| %q | 何も出力しない | × |
| %% | %文字 | × |

「セッションのみ」というのは、各クライアントからの接続などで利用するセッションプロセス経由でのアクセスのときのみに出力されるものを示します。それ以外は、バックグラウンドで動く各プロセスからの出力でも利用可能な項目になります。

## 5.4. MySQL の場合

MySQL の場合、DBMS 上に出力する場合と、ファイルに出力する場合とで出力される項目が異なります。

DBMS 上に出力する場合は、general_log テーブルに以下のような項目を持った情報が出力されます。

**表 5-3 MySQL で保存先を TABLE に設定した際に取得可能な監査情報**

| 列 | データ型 | 説明 |
|---|---|---|
| event_time | timestamp | イベント発生時刻 |
| user_host | mediumtext | ホスト名 |
| thread_id | int(11) | スレッド ID |
| server_id | int(10) unsigned | サーバ ID |
| command_type | varchar(64) | コマンド種別 |
| argument | mediumtext | コマンド引数 |

ファイルに出力する場合は、時刻、スレッド ID、コマンド種別、引数が出力されます。

全てのレコード(行)にユーザの情報が含まれていないため、スレッド ID を利用して過去を辿ってそのスレッド ID でログインしたレコードをみつけないと、ログイン(Connect)したレコード以外のオペレーションを誰がどこから実行したのか特定することができません。

# 6. 監査ログの保全

各 DBMS によって出力された監査ログは、以下の 2 つに分類できます。

- DBMS での管理
- ファイルシステム上のファイル

これらは、監査のために取得したものですので、勝手に改竄されてしまっては正しく監査を行うことができなくなってしまいます。また、監査ログを参照することで DBMS にアクセスしたオペレーションの内容がわかってしまうため、セキュリティ上の脅威となってしまうことが考えられます。

そこで、取得されたこれらの監査ログを、改竄や第三者による盗聴から守ることが必要となります。

## 6.1. Oracle Database の場合

Oracle Database の場合は、監査ログは全て DBMS 上の特殊なテーブルに保存されます。
これらのテーブルは、権限を与えない限り、通常のユーザは参照することができません。

デフォルトでは、システム管理権限を持っている SYS/SYSTEM ユーザのみが参照することができます。

## 6.2. PPAS の場合

PPAS 独自の監査機能の場合は、通常のファイルシステム上にファイルとして出力されます。

出力されるディレクトリやファイルのパーミッションは、デフォルトだと PPAS の管理用ユーザである「enterprisedb」ユーザのみがアクセス可となっており、OS 上の「enterprisedb」ユーザと OS の「root」ユーザのみがアクセスすることができます。
具体的には、所有者と所有グループの両方が「enterprisedb」となっていて、ディレクトリのパーミッションが 700、ファイルのパーミッションが 600 になっています。

## 6.3. PostgreSQL の場合

PostgreSQL のログ出力機能を利用する場合は、通常のファイルシステム上にファイルとして出力されます。

出力されるディレクトリやファイルのパーミッションは、デフォルトだと PostgreSQL の管理ユーザ「postgres」ユーザ[1](PPAS の場合は「enterprisedb」ユーザ)のみがアクセス可となっており、OS 上の「postgres」ユーザと、OS の「root」ユーザのみがアクセスすることができます。
具体的には、所有者と所有グループの両方が「postgres」となっていて、ディレクトリのパーミッションが 700、ファイルのパーミッションが 600 になっています。

## 6.4. MySQL の場合

MySQL の場合は、DBMS 上と通常のファイルシステムのどちらかまたは両方に出力することができます。

DBMS 上では、DBMS のサーバプロセスからしか書き込めない特殊なテーブル上に保存されているため、MySQL 上の root ユーザでも特定のレコードのみを削除したり更新したりすることができないようになっています。

通常のファイルに出力されたものに関しては、所有者と所有グループの両方が「mysql」となっていて、ファイルのパーミッションが 600 になっています。

[1] PostgreSQL の RPM からインストールした場合。

# 7. 監査ログの運用

取得した監査ログは、何もしないとどんどんとサイズが大きくなってしまいますので、定期的にバックアップして退避し、不要になったログをサーバ上から削除するような運用を行う必要があります。

対象のシステムにもよりますが、対象のシステムが準拠すべき法令や基準に合わせて保管期間も考慮しておく必要があります。金融庁の「財務報告に係る内部統制の評価及び監査に関する実施基準」では、保存期間として 5 年程度が考えられるとなっています。

DB サーバ上にログを保管したままの運用も考えられますが､ハードディスクの容量やサーバへの負荷を考慮して、DB サーバ上で保管しておくことが困難である場合は、ログインの制限、データの暗号化などのセキュリティ対策を施したサーバにログを転送して保管することも検討が必要です。

## 7.1. Oracle Database の場合

Oracle Database 11gR2 からは、この監査ログ管理用のパッケージが用意されています。
DBMS_AUDIT_MGMT というパッケージで、これを利用することによって、古いログのアーカイブを取得し、アーカイブを取得した時点までのログを削除することが可能となっています。詳細は、「Oracle Database セキュリティ・ガイド 11g リリース 2（11.2） B56285-02」などをご参照下さい。

## 7.2. PPAS の場合

PPAS の場合は、通常のテキストファイルとして監査ログが出力されますので、これを安全に必要な期間保存しておくために、暗号化や圧縮などの処理を行って保管するようにします。

## 7.3. PostgreSQL の場合

PostgreSQL の場合も、通常のテキストファイルとしてログが出力されますので、これを安全に必要な期間保存しておくために、暗号化や圧縮などの処理を行って保管するようにします。

## 7.4. MySQL の場合

MySQL の場合、DB 上に保存されたログは、Oracle Database のような管理ツールは用意されていません。しかも、期限を区切って古いものを削除することもできません。
古い情報のアーカイブを取得してバックアップし、古い情報を削除したい場合は、テーブルごと入れ替えを行います。

具体的には以下のような手順で行います。

① 同じスキーマの空のテーブルを作成
例：
mysql> USE mysql;
mysql> CREATE TABLE IF NOT EXISTS general_log2 LIKE general_log;
② 現在のテーブルのテーブル名をバックアップ用のテーブル名に変更し、同時に新しく作成しておいたテーブルを監査ログ保存用のテーブル名に変更して入れ替え
例：
mysql> RENAME TABLE general_log TO general_log_backup, general_log2 TO general_log;
③ バックアップ用のテーブルをダンプしてバックアップ
④ バックアップ用のテーブルを削除

ファイルにログ出力した場合は、ファイル名を変更して、mysqladmin flush-logs を実行することでファイルが切り替えられ、元のファイル名で新しく出力が開始されるようになります。
ファイルとして出力されたログは、通常のテキストファイルですので、これを安全に必要な期間保存しておくために、暗号化や圧縮などの処理を行って保管するようにします。

# 8. まとめ

全ての DBMS において、最低限、「いつ、誰が、どこから、どのような操作を行ったか」といった情報を取得して、DB 上もしくはファイルとして保存するよう設定できることがわかりました。
ただし、設定して取得はできるのですが、いくつか留意しなければならない点がありますので、「監査のしやすさ」「ログの容量」「ログのセキュリティ対策」という視点で整理して以下に記述します。

## 8.1. 監査の容易性

監査ログを取得後、実際に監査を行う際には注意が必要です。
特に、MySQL の場合は、監査ログの全てのレコードには「誰が」「どこから」という情報が含まれていないため、行われた操作に対して誰がどこから行ったかを特定するためには、過去のログも辿ることが必要になります。つまり、監査作業を行うために他の DBMS を利用する場合よりも手間がかかります。また、MySQL の場合は、監査ログ内に検索や更新の要求に対してその操作が成功したのか失敗したのかを示す SQL コード相当の情報が含まれないため、正確にその操作が反映されたのかを判断できないのも課題です。

Oracle Database や MySQL で保存先として DB を選択していた場合は、大量のログから目的のキーワードを含む操作を SQL 文で指定して取得することができるため、比較的容易に目的のレコードを抽出して監査作業を行えると思います。
PostgreSQL や PPAS を利用していた場合には、CSV 形式で出力しておくことで、別途、分析用の DB にログをロードして同様の分析を行うことができます。したがって、テキストファイルに出力する場合には、CSV 形式を利用しておくと、監査作業の負荷を軽減できるでしょう。

## 8.2. ログの容量

監査ログを取得する際の問題として監査ログのサイズがあげられます。

Oracle Database で取得できるもの全てを網羅するためには、PPAS、PostgreSQL、MySQL では、ほとんど全ての操作ログを出力するような設定にしなければなりません。そうしてしまうと、監査ログのサイズは大きくなってしまいますので、ハードディスクの容量を監査ログ用に余裕を持って確保しておくことが必要になります。また、それだけハードディスクへの入出力が増加すると、DBMS のパフォーマンスへの影響も懸念されます。

Oracle Database ではファイングレイン監査の機能などを利用して細かな単位での監査対象の絞込みが可能なのですが、他の DBMS ではログを保存する対象を本当に必要なものだけに限定することができません。そうすると、アプリケーションで利用する作業用テーブルへのアクセスなど、監査する必要が無いようなデー

タベースやテーブルに対する多数のクエリの情報も一緒に保存することとなってしまいます。それによって、本来のアプリケーションのための DBMS としての処理以外の処理が増えたり、ハードディスクなどの容量を消費してしまったりする危険性があります。

この監査ログ出力に関する各 RDBMS の機能比較をまとめると次の表のようになります。

**表 8-1 機能比較**

| | | Oracle | PPAS | PostgreSQL | MySQL |
|---|---|---|---|---|---|
| 保存先 | ファイル | ○ | ○ | ○ | ○ |
| | DB | ○ | × | × | ○ |
| 監査指定の反映 | 出力先指定 | 起動時に指定 | 起動時に指定 | 起動時に指定 | 起動時に指定 |
| | 監査対象指定 | 動的に変更可 | 起動時に指定 | 動的に変更可 | 起動時に指定 |
| 監査対象 | 接続／切断 | ○ | ○ | ○ | ○ |
| | データ操作 | ○ | ○ | ○ | ○ |
| | テーブル操作 | ○ | ○ | ○ | ○ |
| | 検索 | ○ | ○ | ○ | ○ |
| | エラー | ○ | ○ | ○ | ○ |
| 取得単位 | システム全体 | ○ | ○ | ○ | ○ |
| | ユーザ | ○ | × | ○ | × |
| | データベース | ○ | × | ○ | × |
| | テーブル | ○ | × | × | × |
| | 対象条件有り | ○ | × | × | × |

ここで PPAS という列で挙げているのは、PPAS が独自に持っている監査機能を利用した場合であって、PostgreSQL の機能を利用すれば、PostgreSQL の列に記載した対応が可能です。

対象となるシステムの業種や提供するサービス内容に合わせて、各業界での監査で必要になる情報を確認し、監査ログとして出力する対象をしぼることで、監査ログの量を抑えることができます。

## 8.3. ログのセキュリティ対策

監査ログの出力先としてファイル出力を選択した場合は、通常のファイルシステム上に出力されるものであるため、OS の特権アカウントを利用して盗み見たり改竄されたりしてしまう可能性があります。ログ出力先のディレクトリやファイルのパーミッションにも注意が必要です。出力されたログファイルの暗号化や DB サーバマシンへのログインを極力制限するなどのセキュリティ対策も検討してください。

# 9. 今後の課題

このレポートを記述している時点では、監査ログを出力するように設定した際のパフォーマンスに対する影響度は測定できていません。

もし、ログを保存しておくのに十分なハードディスクが用意できたとしても、DBMS としての処理パフォーマンスへの影響が大きい場合は、さらに監査対象をしぼるなどの対策が必要になることが予想されます。

環境を調達して、デフォルトの設定の場合と監査ログを取得する場合とを比較して、どの程度のパフォーマンス低下が発生してしまうかを確認し、その結果を改めて公開する予定です。

# 10. 付録

## 10.1. 参考資料

- Oracle Database 11gR2 マニュアル http://download.oracle.com/docs/cd/E16338_01/index.htm
- Postgres Plus Advanced Server マニュアル http://www.enterprisedb.com/documentation/english
- PostgreSQL マニュアル(English) http://www.postgresql.org/docs/9.0/static/index.html
- PostgreSQL マニュアル(日本語) http://www.postgresql.jp/document/pg904doc/index.html
- MySQL マニュアル http://dev.mysql.com/doc/refman/5.5/en/index.html
- 財務報告に係る内部統制の評価及び監査の基準並びに財務報告に係る内部統制の評価及び監査に関する実施基準の改訂について（意見書） 平成２３年３月３０日 金融庁
  http://www.fsa.go.jp/singi/singi_kigyou/tosin/20110330.html
- 情報セキュリティ管理基準 (平成 20 年改正版) 平成 20 年経済産業省告示第 246 号
  http://www.meti.go.jp/policy/netsecurity/audit.htm

## 10.2. 関連サイト

- Postgres Plus Advanced Server http://www.enterprisedb.com/
- PostgreSQL http://www.postgresql.org/
- 日本 PostgreSQL ユーザ会 http://www.postgresql.jp/
- MySQL http://www-jp.mysql.com/