レプリケーション

# レプリケーションとは

レプリケーションとは、レプリカ（複製）の名のとおりデータをインスタンスから別のインスタンスに複製する仕組みのことです。可用性向上や負荷分散の目的で利用されます。

MySQLでは元となるデータベースを「**マスタ**」、複製先のデータベースを「**スレーブ**」と呼びます。

そして、MySQLではバイナリログを利用してレプリケーションを実現します。

各スレーブはバイナリログをマスタから取得し、スレーブの「**リレーログ**」に保存します。

リレーログをスレーブ側のデータベースに適用することでマスタと同じ状態にします。

この作業を断続的に実施することでデータの状態を保ちます。

　　　マスタ

MySQL

server\_id=1

データ

ファイル

バイナリ

ログ

　　スレーブ１

MySQL

server\_id = 2

データ

ファイル

バイナリ

ログ

リレー

ログ

　　スレーブ２

MySQL

server\_id = 3

データ

ファイル

バイナリ

ログ

リレー

ログ

レプリケーションでは複数のサーバーでMySQLを利用するため、各MySQLサーバーを識別する必要があります。その為、MySQLのserver\_idは全て異なる値を設定します。

server\_idはMySQL8.0からデフォルトで1が指定されています。

# バイナリログ転送方式でのレプリケーション

MySQLのレプリケーション実装方法には大きく２つあります。バイナリログ転送を利用した方式とグループレプリケーション方式です。

先ほどのイメージ図はバイナリログ転送方式です。バイナリログ転送方式では、マスタとスレーブの同期処理を行うタイミングがあり「バイナリログのファイル名やポジション」を使用する方法と「GTID(Global Transaction Identifier)」を利用する方法があります。

授業では、ベーシックな「**バイナリログ転送方式でバイナリログのポジションを利用して同期**」する方法でレプリケーションの学習を進めます。

# スレーブの処理について

　マスタ

MySQL

バイナリログ

　スレーブ1

MySQL

　スレーブ２

MySQL

SELECT

UPDATEなど

UPDATEなど

スレーブはマスタのレプリカの為、**スレーブに対して更新処理を行ってはいけません**。

ただし、**バイナリログに記録されない参照系クエリはスレーブで実行しても大丈夫**です。

その為、更新より参照が多いシステムでは、スレーブを複数作成して参照処理をスレーブで

行うことで負荷分散したり、データ分析処理などの参照に時間が掛かる処理などをスレーブで行うことがあります。

なお、参照処理はマスタかスレーブ、更新処理はマスタで処理を行うことなどの振り分けはMySQLでは行いません。利用側のクライアントアプリケーションで適切に設定を行う必要があります。

また、一部ロードバランサやミドルウェアでこの振り分けを行う機能がある場合は、クライアントアプリケーションでは無くロードバランサ側で振り分けを行うことも可能です。

ロードバランサ/DNSなど

業務システムなど

　マスタ

MySQL

バイナリログ

　スレーブ1

MySQL

 リレー

ログ

更新系

トランザクション

参照系

トランザクション

ハンズオン　レプリケーションの準備

本来、レプリケーションの構築を行う場合、マスタデータベース以外のサーバーが必要になります。授業では、各自ノートPCのVirtual Boxに仮想OS(SentOS)を立てることで

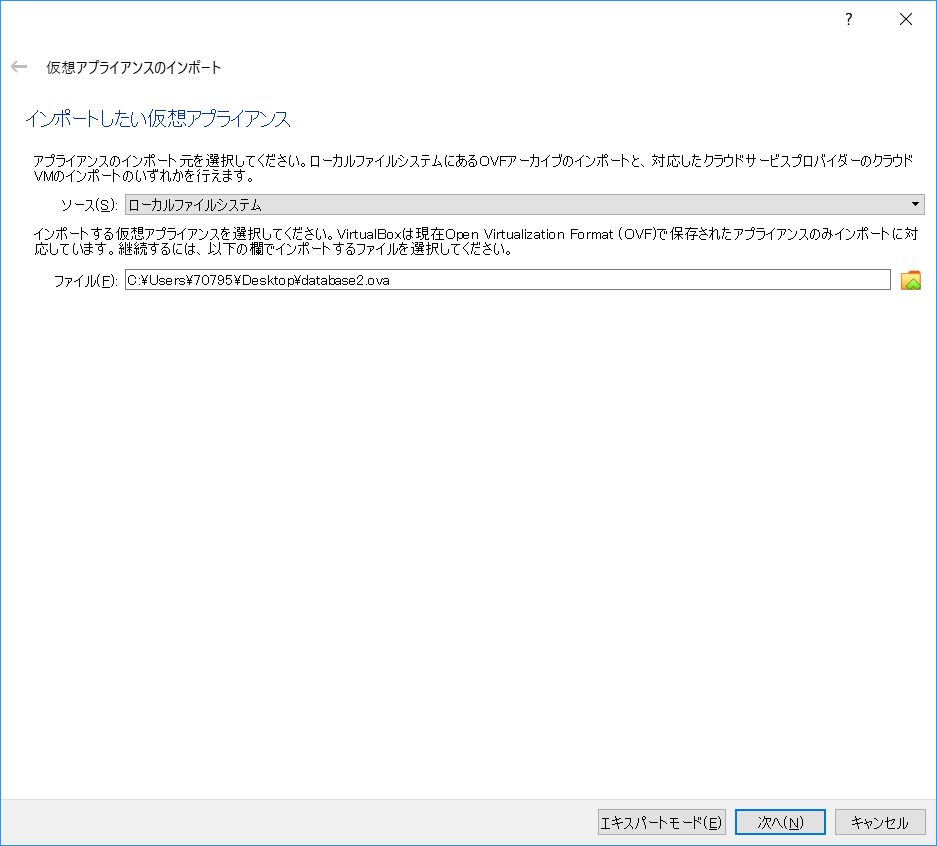
スレーブ用のサーバーを２つ用意して学習を進めていきます。

※事前準備として、配布するdatabase2.ovaをノートPCにダウンロードしておくこと。

１． Virtual Boxを起動して「ファイル - 仮想アプライアンスのインポート」を選択

　　ファイルにダウンロードしたdatabase2.ovaを指定して「次へ」

　　そのまま設定を変更せずに「インポート」を行う。



・学習環境イメージ

windows

Virtual Box

ノートPC

MySQL

Linux

MySQL

Linux

MySQL

Linux

　　ユーザ：root

パスワード：osroot

MySQL

ユーザ：root

パスワード:#RootEcc144

仮想PC情報

※DB\_SentOs7、DB2\_SentOs7共に同じ

レプリケーション間でserver\_idは重複してはいけません。

ゲストOSのMySQLのserver\_idの変更を行います。

２．DB\_SentOS7を起動させて、ゲストOSにログインする。

localhost login：osroot

Password：osroot

３．ゲストOSのMySQLを停止させる

　　systemctl stop mysqld.service

４．ゲストOSのMySQL環境ファイルにserver\_idの設定を追記する

　　nano /etc/my.cnf

【エディタが開くのでファイルの最後に以下のパラメータを追記】

server\_id=2

【ctrl+oで上書き、ctrl+xでエディタを閉じる】

５．ゲストOSをシャットダウンする

shutdown -h now

６．DB2\_SentOS7にも、同様の作業を行う。

**※ただし、server\_idは3で設定すること！！**

次に、レプリケーションを行う場合には、レプリケーション接続用のユーザをマスタ側に作成を行います。なお、**レプリケーション接続用ユーザはREPLICATION SLAVE権限だけを持ったユーザ**を作成します。

７．ホストOS(windows)のMySQLに管理者権限でログインします。

mysql -u root -p

８．レプリケーション接続用のユーザreplを作成します。

CREATE USER repl@'%'

IDENTIFIED WITH MYSQL\_NATIVE\_PASSWORD BY 'replpass';

９．REPLICATION SLAVE権限を付与する

　　GRANT REPLICATION SLAVE ON \*.\* TO repl@'%';

FLUSH PRIVILEGES;

# 検証シナリオについて

今回のレプリケーションはバイナリログ転送方式で行うので、全体バックアップを取得後にデータの更新を行った後に、スレーブ側の設定を行います。

スレーブ

マスタ

バックアップ

ログ

ログ

ログ

レプリケーション開始

ログ

レプリケーション設定

ハンズオン　レプリケーション構築

１．【マスタ】mysqldumpで全体フルバックアップのダンプファイルを作成する

mysqldump -u root -p --all-databases --master-data > repldump.sql;

２．【マスタ】ダンプファイルをテキストで開いて、MASTER\_LOG\_FILEと

　　　　　　MASTER\_LOG\_POSを確認する。

３．【マスタ】dbuserでstudbにログインしてデータを登録する。

INSERT INTO MATERIAL VALUES ('00025', '明太子', '福岡', 80);

INSERT INTO MATERIAL VALUES ('00026', 'マヨネーズ', NULL, 15);

INSERT INTO MATERIAL VALUES ('00027', 'もち', NULL, 50);

４．【スレーブ】Viratul BoxでDB\_SentOS7を起動させる

５．【スレーブ】ゲストOSのIPアドレスを表示する

ifconfig

６．【スレーブ】バックアップファイルをゲストOSにコピーするため

　　　　　　　　TeraTermでゲストOSに接続する

７．【スレーブ】repldump.sqlをTeraTermにドラッグアンドドロップする

８．【スレーブ】ゲストOSのデータベースにダンプファイルをリストアする

※パスワード：#RootEcc144

　　mysql -u root -p < repldump.sql

９．【スレーブ】再度mysqlにログインをしてマスタサーバの指定を行う

　　mysql -u root -p

【ログイン後】

　　 CHANGE MASTER TO

MASTER\_HOST="**127.0.0.1**",

MASTER\_USER="repl",

MASTER\_PASSWORD="replpass",

MASTER\_LOG\_FILE="**binlog.000221**",

MASTER\_LOG\_POS=**2115** ;

１０．【スレーブ】レプリケーションを開始する

START SLAVE;

１１．【スレーブ】マスタの情報が反映されているか確認する。

　　SELECT \* FROM STUDB.MATERIAL;

１２．【マスタ】新たにデータを登録する

　　INSERT INTO MATERIAL VALUES ('00028', '韓国のり', '韓国', 20);

COMMIT;

１３．【スレーブ】マスタの情報が反映されているか確認する。

　　SELECT \* FROM STUDB.MATERIAL;

# レプリケーションのトラブルシューティング

レプリケーションを構築していると、マスタサーバに障害が発生したときに

前回学習したリカバリ処理を行う代わりに、スレーブの一つをマスタサーバに昇格させることで対応することが可能です。

マスタ

スレーブ

マスタ

ログ

ログ

ログ

スレーブをマスタに切替

ハンズオン　マスタの切り替えとレプリケーション再構築

１．【マスタ】障害が発生した想定で、データベースを停止する

**～～～　これ以降は、課題の問２で実施する内容です。～～～**

２．【スレーブ】スレーブ機能を停止して、スレーブ情報を削除する

　　STOP　SLAVE;

　　RESET SLAVE ALL;

３．【スレーブ(新マスタ)】ログファイル名とログポジションを確認する

　　SHOW MASTER STATUS;

※他のスレーブが存在する場合は、ここで確認したログファイル、ログポジションで

スレーブ設定をやり直すことでマスタの切り替えが可能となる。