

PostgreSQL エンタープライズ・コンソーシアム 技術部会 WG#2

異種 DB 間連携調査編

製作者
担当企業名
サイオステクノロジー株式会社

改訂履歴

版	改訂日	変更内容
1.0	2013/01/12	新規作成

ライセンス



本作品は CC-BY ライセンスによって許諾されています。

ライセンスの内容を知りたい方は <http://creativecommons.org/licenses/by/2.1/jp/> でご確認ください。

文書の内容、表記に関する誤り、ご要望、感想等につきましては、PGECcons のサイトを通じてお寄せいただきますようお願いいたします。

サイト URL <https://www.pgecons.org/contact/>

Eclipse は、Eclipse Foundation Inc の米国、およびその他の国における商標もしくは登録商標です。

IBM および DB2 は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Intel、インテルおよび Xeon は、米国およびその他の国における Intel Corporation の商標です。

Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Red Hat および Shadowman logo は、米国およびその他の国における Red Hat, Inc. の商標または登録商標です。

Microsoft、Windows Server、SQL Server、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

MySQL は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

Oracle は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

PostgreSQL は、PostgreSQL Community Association of Canada のカナダにおける登録商標およびその他の国における商標です。

Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です。

TPC、TPC Benchmark、TPC-C、TPC-E、tpmC、TPC-H、QphH は米国 Transaction Processing Performance Council の商標です

その他、本資料に記載されている社名及び商品名はそれぞれ各社が商標または登録商標として使用している場合があります。

はじめに

■本資料の概要と目的

今日では事業や技術の目まぐるしい変化に伴い、サーバ上のデータも統合や連携を効率良く実施する機会や必要が多々あります。また、ここ数年でクラウド環境の利用も進んだ事により、クラウドとオンプレミス環境間にあるデータ連携や、災害対策として遠隔地サーバとのデータ連携など、その需要も多様です。本書では、このような需要の背景を基に異種 DBMS と PostgreSQL の連携を考察し、利用できるソフトウェアや、実際のソフトウェアの使用手順を確認してみます。また、作業を実施する中で制約事項やシステム設計時の注意点も確認する事で、異種 DBMS 連携時に考慮すべき点を理解していきます。

■資料で扱う用語の定義

No.	用語	解説
1	レプリケーション	複数のサーバ間でデータベースのデータやオブジェクトを共有する仕組みを意味します。
2	レプリカ	レプリケーションによって複製されたデータベースサーバを指します。
3	DWH(データウェアハウス)	大規模データベースを意味します。
4	オンプレミス	自社で運用管理するシステムを意味します。
5	Oracle	データベース管理システムの Oracle Database を指します。
6	SQL Server	データベース管理システムの Microsoft SQL Server を指します。
7	異種 DBMS	PostgreSQL および EnterpriseDB 社の Postgres Plus 製品以外のデータベースソフトウェアを意味します。

目次

1.異種 DBMS と PostgreSQL の連携について.....	5
1.1.異種 DBMS と PostgreSQL を連携する目的.....	5
1.2.異種 DBMS と PostgreSQL の連携を実現する際の課題.....	5
2.データベース連携ソフトウェアの紹介.....	6
2.1.InfoFrame DataCoordinator.....	6
2.2.DBMoto.....	6
2.3.DataSpider Servista.....	6
2.4.xDB Replication Server.....	7
3.異種 DBMS 間連携のトライアル.....	8
3.1.ツールのインストール.....	9
3.2.コネクタの準備.....	9
3.3.パブリケーションサーバとサブスクリプションサーバについて.....	9
3.4.パブリケーションサーバとサブスクリプションサーバの登録.....	10
3.5.データの同期対象 DBMS と PostgreSQL の登録.....	10
3.6.複製するオブジェクトの選択.....	11
3.7.レプリケーションの設定.....	11
3.8.その他の設定.....	12
4.レプリケーションシステムのデザインについて.....	13
4.1.データを連携する上での注意事項.....	13
4.2.レプリケーションシステムの検討事項.....	13
4.3.レプリケーションオブジェクトの制限.....	14
4.4.可能な DBMS 連携構成例.....	15
5.まとめ.....	17

1. 異種 DBMS と PostgreSQL の連携について

1.1. 異種 DBMS と PostgreSQL を連携する目的

企業ではビジネスの変化や情報処理技術の急激な進歩により、当初のシステム設計とは異なる要件やコストでシステム拡張が必要となる場合があります。しかし、これまで築いたアプリケーションやシステムの急な変更は難しく、価値のある情報は既存システムにある場合がほとんどです。このような状況から、今日では既存の DBMS を生かしつつ、オープンソースソフトウェアである PostgreSQL でシステムを拡張する機会が増えてきています。異種 DBMS と PostgreSQL のデータベースを連携させる目的や機会は様々ですが、連携させる目的は概ね以下のようなケースがあります。

- データベース移行に依る期間限定の連携
例: 事業統合などに伴うデータの統合や、開発サーバへのデータ移行や同期。
- 計画的なシステム構成変更依る連携
例: 事業拡大に伴うマスター DB のレプリカ増設。
- 負荷分散に依る連携
例: マスター DB の負荷分散の為のレプリカの増設、トランザクションの軽減。
- 災害対策
例: 災害対策に備えて遠隔地にマスター DB の複製を確保。
- DWH 構築
例: 大量のデータ分析の為のデータ統合など。

1.2. 異種 DBMS と PostgreSQL の連携を実現する際の課題

データベースソフトウェアは、開発メーカーによってデータベースエンジンの構造やデータの表現方法が異なり、データ操作方法のルールなども異なっています。その為、異種 DBMS と PostgreSQL 間でデータの連携を取るには、これらの差異を吸収してデータを連携させる仕組みが必要になります。

このようなデータ連携の仕組みを開発・構築するには、各データベースソフトウェアへの深い理解と高度な技術が必要であり、自社で開発するには敷居が高く、多くの時間とコストが掛かります。また、その後のメンテナンスやサポートも考慮すると、先の見えない投資に成りかねません。データを連携させる仕組みは以前から提供されていましたが、容易に構築・管理する事を実現するには、個々の需要に見合うエンタープライズ向けの既製品を利用した方が得策となる場合が多いと言えます。

異種 DBMS と PostgreSQL を連携させるソフトウェアは、SymmetricDS のようなオープンソースソフトウェアとして無償で提供されているものもありますが、本書ではエンタープライズ環境において迅速かつ確実に要件が実現できる商用ソフトウェアについて紹介をします。

2. データベース連携ソフトウェアの紹介

この項では異種 DBMS と PostgreSQL のデータ連携が可能な商用ソフトウェアを紹介します。各ソフトウェアの詳細については、各開発ベンダーが提供している情報をあわせてご確認ください。

表 1: 商用ソフトウェア一覧¹

項番	ソフトウェア名	ベンダー
2.1.	InfoFrame DataCoordinator	日本電気株式会社
2.2.	DBMoto	株式会社クライム
2.3.	DataSpider Servista	株式会社アプレッソ
2.4.	xDB Replication Server	EnterpriseDB

2.1. InfoFrame DataCoordinator

製品名: InfoFrame DataCoordinator
URL: <http://www.nec.co.jp/pfsoft/datacoordinator/>
ベンダー: 日本電気株式会社

特徴:

- GUI ツールのみで直感的にレプリケーションの設定などが可能
- 直感的に操作が可能な Designer GUI ツールを提供
- DataCoordinator サーバに Windows Server が必要だが、ACCOS-4 のメインフレームや Linux の対応がある

2.2. DBMoto

製品名: DBMoto
URL: <http://www.climb.co.jp/soft/dbmoto/>
ベンダー: 株式会社クライム

特徴:

- IBM i Series DB2/400(AS/400) との連携に強みがある
- 製品の使用には Windows Server が必要だが、データベースの OS は依存しない

2.3. DataSpider Servista

製品名: DataSpider Servista
URL: <http://dataspider.appresso.com/>
ベンダー: 株式会社アプレッソ

特徴:

- クラウドとオンプレミスの連携が強化されている
- プログラミングなしに GUI で連携設定・操作が可能

¹ 本文に記載されている会社名、製品名は各社の商号または登録商標です。

2.4. xDB Replication Server

製品名: xDB Replication Server

URL: <http://www.enterprisedb.com/products-services-training/products-overview/postgres-plus-solution-pack/xdb-replication-server>

ベンダー: EnterpriseDB

特徴:

- マルチマスターレプリケーションが可能
- Oracle と高い互換性がある
- xDB Replication Server の設定や管理を行う xDB Replication Console という GUI ツールを提供

3. 異種 DBMS 間連携のトライアル

本項では実際に商用ツールの一つを用いて、どのように異種 DBMS と PostgreSQL が連携できるか手順を追った概要を記載します。

今回使用するツールは、EnterpriseDB 社が提供している xDB Replication Console を用います。当該ツールはレプリケーションを実現する xDB Replication Server の設定や管理を行うグラフィカルなインターフェースのツールであり、xDB Replication Server の一つのコンポーネントとして同梱されています。当該ツールの画面を通じて、Oracle, SQL Server から PostgreSQL, EnterpriseDB 社の Postgres Plus 製品へのレプリケーション、また PostgreSQL 間のレプリケーションを設定する事が可能であり、複数のレプリケーション接続設定も容易に実現する事ができます。このツールが含まれている xDB Replication Server は単体で購入する事が可能な為、コミュニティ版の PostgreSQL と組み合わせて使用する事ができます。

なお、このトライアルではツールを使用した異種 DBMS 間の連携過程を確認し、レプリケーション設定がツールによって煩雑な作業なく実現できる事を見ていきます。今回のトライアルでは、複製元の異種 DBMS に Oracle Database 10g Express Edition を使用し、複製先の DBMS には Oracle と互換機能がある EnterpriseDB 社製の Postgres Plus Advanced Server (以後 Postgres と呼称)を使用してテーブルの複製を実施します。



表 3.1 使用ソフトウェア

項目	ソフトウェア名	バージョン
既存の異種 DBMS	Oracle Database 10g Express Edition	10.2.0.1.0
複製先 DBMS	Postgres Plus Advanced Server * Postgres Plus 製品は PostgreSQL をベースに、周辺ツールや生産性の高いコンポーネントをバンドルした EnterpriseDB 社の製品です。	9.2.2.4
複製 GUI ツール	xDB Replication Console * 有償の商用ツールであり、xDB Replication Server を単体で購入すると同梱されている GUI ベースのツールです。Postgres Plus 製品の購入でも利用する事が可能です。	5.0.2

3.1. ツールのインストール

EnterpriseDB 社が提供している Postgres Plus 製品を導入、または xDB Replication Server を単体購入すると、xDB Replication Console というグラフィカルなレプリケーション設定・管理ツールが利用できます。これをインストールして起動した画面は以下です。

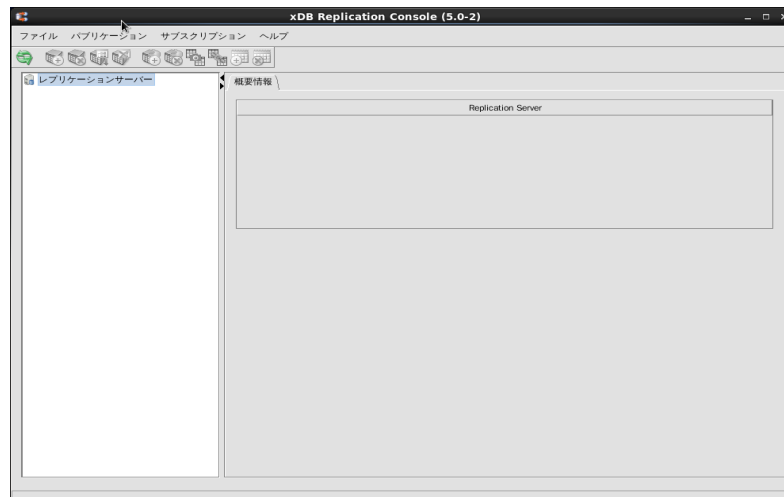


図 3.1: xDB Replication Console 起動画面

3.2. コネクタの準備

xDB Replication Console から異種 DBMS に接続を行う為には、コネクタというファイルを別途用意する必要があります。このファイルは多くの場合、ライセンスの関係で異種 DBMS のベンダーが提供しているものを入手する必要があります。今回は ojdbc14.jar というコネクタファイルをオラクル社の web サイトから入手し、Postgres Plus のホームディレクトリ配下にある ./lib/jdbc というサブディレクトリに配置しておきます。

(コネクタファイルの配置ファイルパスの例)

```
/opt/PostgresPlus/9.2AS/lib/jdbc/ojdbc14.jar
```

3.3. パブリケーションサーバとサブスクリプションサーバについて

xDB Replication 製品では、publish and subscribe という仕組みを用いてデータを複製します。複製元となる DBMS のデータコピーを可能とする側を「パブリケーションサーバ(登録公開サーバ)」と定義し、当該コピーデータを受け取る側を「サブスクリプションサーバ(登録定期サーバ)」と呼称します。パブリケーションサーバとサブスクリプションサーバは、xDB Replication 製品の内部に組み込まれたサーバコンポーネントです。これらのサーバプロセスは、同一ホスト内で動作も可能ですし、異なるホスト間に配置する事も可能です。手順例では、同一ホストで動作をさせています。



図 3.2: xDB Replication Console

3.4. パブリケーションサーバとサブスクリプションサーバの登録

パブリケーションサーバとサブスクリプションサーバを登録すると、以下のような画面になります。パブリケーションサーバにある SMR と MMR は、それぞれ Single Master Replication と Multi Master Replication を意味します。今回のトライアルではこれらの違いの説明は目的と異なる為、詳細を割愛して Single Master Replication 側に実際のデータの複製元となる異種 DBMS の接続情報を登録して作業を進めます。



図 3.3: xDB Replication Console 設定

3.5. データの同期対象 DBMS と PostgreSQL の登録

前述したパブリケーションサーバとサブスクリプションサーバに、実際にレプリケーションを行う複製元と複製先の DBMS の接続情報(IP アドレスやポート番号、接続ユーザ情報等)を登録します。今回は複製元となるパブリケーションサーバに Oracle, 複製先となるサブスクリプションサーバに Postgres を登録します。登録し終えた画面は、以下のような表示になります。

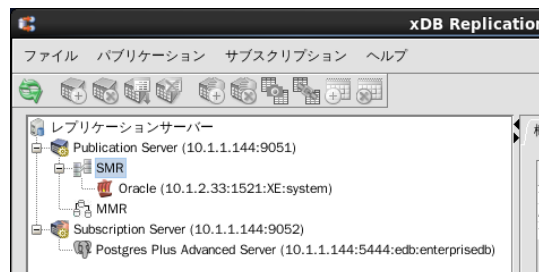


図 3.4: 複製元と複製先サーバの設定

3.6. 複製するオブジェクトの選択

複製元となる異種 DBMS において、どのテーブルを複製するか決定します。この例では、Oracle にサンプルとして含まれている HR スキーマのテーブルやビューをいくつかレプリケーションする設定を行います。複製対象として選択したオブジェクト(テーブルやビュー)は、以下のようにパブリケーションサーバに登録した Oracle のツリーとして表示され、この複製対象データの定義は、パブリケーションと呼称されます。作成したパブリケーションは、識別する為の名前を任意に付ける事が可能であり、このトライアルのケースでは ora.pub と設定しています。

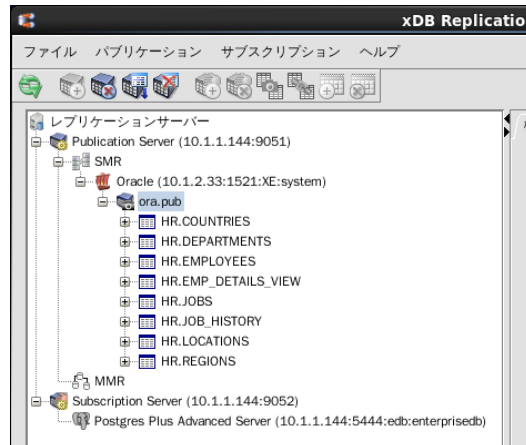


図 3.5: 複製するオブジェクトの設定

3.7. レプリケーションの設定

パブリケーションサーバ側の設定を終えたら、サブスクリプションサーバ側にある Postgres にサブスクリプションの作成を行います。サブスクリプションの作成では、前述の手順で作成した同期させるデータ定義(パブリケーション)を選択し、同期するデータ内容を決定します。作成したサブスクリプションにも識別の為の名前を任意に付ける事が可能であり、このトライアルでは ora.sub という名前を設定しています。レプリケーションの基本設定は以上で完了です。以下のように Postgres 側に複製されるオブジェクトのツリーが表示され、Oracle 側で生じたテーブルデータの更新内容を Postgres 側のテーブルに反映して同期する事が可能となります。

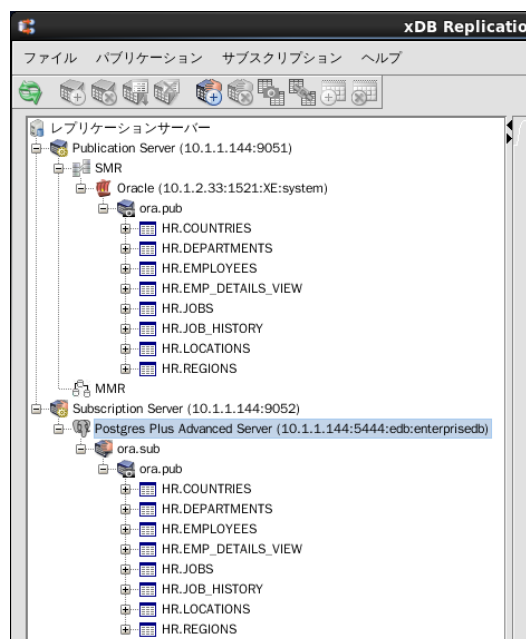


図 3.6: レプリケーション設定

3.8. その他の設定

xDB Replication Console ではデータの同期モードの設定が 2 種類用意されており、行の更新毎にデータを同期させる同期モードと、テーブルの全てのデータを毎回更新する snapshot モードがあります。snapshot モードでは、同期させる毎に複製先のデータは TRUNCATE され、複製元のテーブルデータが全てリロードされます。今回のトライアルでは snapshot モードを使用しています。

なお、xDB Replication Console では複製対象のデータ型やオブジェクトによって設定できる同期モードに制限があります。例えば、複製元のデータにビューが含まれる場合は、snapshot モードしか設定をする事ができません。また、使用する同期モードによって、テーブルデータ操作の制限内容も異なります。当該内容については、「4. レプリケーションシステムのデザインについて」という後の項目で詳細を解説します。

その他、データを同期するタイミングは、下記ウィザードのようにスケジューリングの設定で変更をすることができます。同期を実行する周期の最小値は、常時を選択した後に秒単位で調整する事ができ、最小値は 1 秒となります。

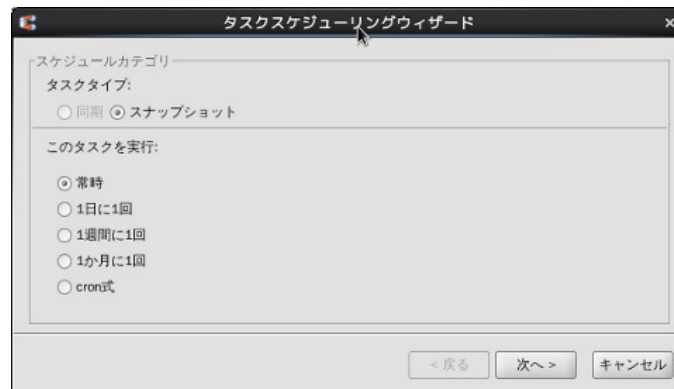


図 3.7: タスクスケジューリングウィザード

4. レプリケーションシステムのデザインについて

異種 DBMS 間の連携で使用するツールの特性によって、制限事項やそれらを踏まえたレプリケーションシステムのデザインは異なるものと考えられますが、本項ではケーススタディとして xDB Replication Console を用いた時における注意事項や考慮すべき点について見ていきます。

4.1. データを連携する上での注意事項

xDB Replication Console の snapshot モードでは、一度レプリケーションを作成したら複製元にあるテーブルの定義やデータの制約は直接変更してはいけません。このような操作を行うと、レプリケーションプロセスにおいて多くのエラーが発生します。これはレプリケーション設定時に、パブリケーションテーブルに関連するトリガーが作成される為です。その為、複製元のテーブル定義を直接変更してしまうと、当該トリガーは機能しなくなります。変更してはいけない具体的なテーブル定義やデータの制約は以下の通りです。

- カラムの追加・削除
- カラム名の変更
- カラムのデータタイプの変更
- カラム長の変更
- NOT NULL 制約の変更
- UNIQUE 制約の変更
- CHECK 制約の変更

もしこのような変更が必要な場合は、一度パブリケーションの設定を削除し、再作成する必要があります。なお、当該制約事項は、使用する同期モードによって条件が異なります。

4.2. レプリケーションシステムの検討事項

xDB Replication Console を使用してレプリケーションシステムを構築する場合は、次の点を考慮する必要があります。

- snapshot モード以外では、パブリケーションサーバ上のテーブルはプライマリーキーを有していなければなりません。
- 複製元の DBMS のテーブル定義(DDL)は、システム運用後に変更が生じないように、レプリケーションシステムを構成する前に十分に検討されたものでなければなりません。同期モードによっては、後から DDL の変更を行えるものもありますが、DDL はシステム運用後に頻繁に変更されるものであってはなりません。
- VIEW は snapshot モードでのみ複製する事ができます。複製先のデータベースにおいては、当該 VIEW はひとつのテーブルとして複製されます。
- サブスクリプションサーバ上に複製されたテーブルは、さらに複製可能なテーブルとして、別途パブリケーションサーバに登録する事ができます。このような特性を生かす事に依って、カスケードレプリケーションを実現する事もできます。具体的な構成例は、後の項目にある ”4.4 可能な DBMS 連携構成例” で示します。
- xDB Replication Console では、双方向にデータの複製の設定を行う事が可能です。具体的な構成例は、後の項目にある ”4.4 可能な DBMS 連携構成例” で示します。

4.3. レプリケーションオブジェクトの制限

レプリケーションで使用するツールに依存して特性は異なりますが、xDB Replication Console で複製できるオブジェクトや制限について記載します。

xDB Replication Console で複製できるオブジェクトは次のようなものです。

- テーブル
- ビュー (snapshot モードでのみ可能)
- プライマリキー
- NOT NULL 制約
- UNIQUE 制約
- CHECK 制約
- インデックス

Oracle テーブルにおいて次のようなデータ型が含まれているものは、xDB Replication Console で複製する事ができません。

- BFILE
- BINARY_DOUBLE
- BINARY_FLOAT
- MLSLABEL
- XMLTYPE

Oracle テーブルにおいて次のようなデータ型は、snapshot モードにおいてのみ複製する事ができます。行更新毎に複製する同期モードでは、これらのオブジェクトを複製する事はできません。

- BLOB
- CLOB
- LONG
- LONG RAW
- NCLOB
- RAW

SQL Server テーブルにおいては、次に挙げるオブジェクトは xDB Replication Console で複製する事ができません。

- GEOGRAPHY
- GEOMETRY
- SQL_VARIANT

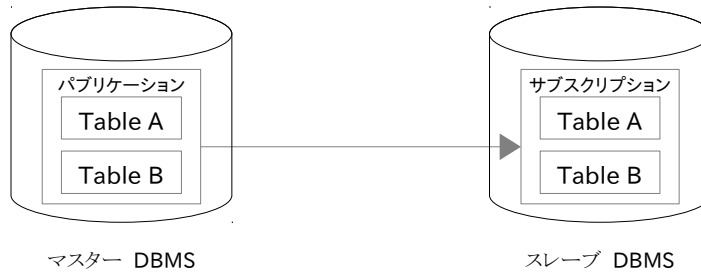
SQL Server テーブルにおいて次のようなデータ型は、snapshot モードにおいてのみ複製する事ができます。行更新毎に複製する同期モードでは、これらのオブジェクトを複製する事はできません。

- BINARY
- IMAGE
- NTEXT
- NVARCHAR(max)
- TEXT
- TIMESTAMP
- VARBINARY
- VARBINARY(max)

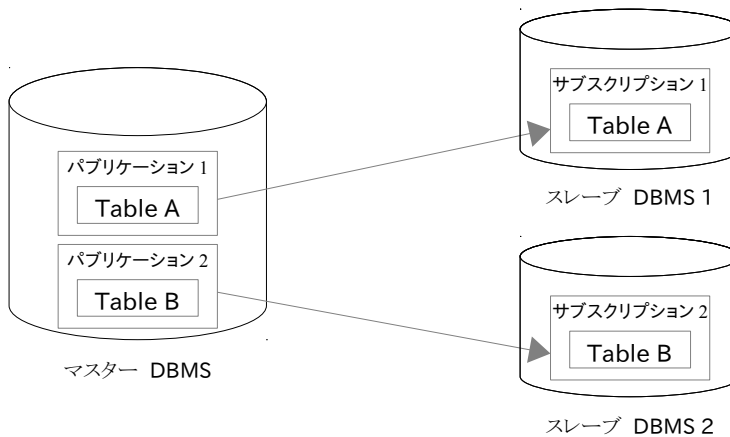
4.4. 可能な DBMS 連携構成例

xDB Replication Server および xDB Replication Console を用いて設定できる代表的なレプリケーションの構成例を次にいくつか挙げます。レプリケーションの要件に応じた構成例をご参考ください。

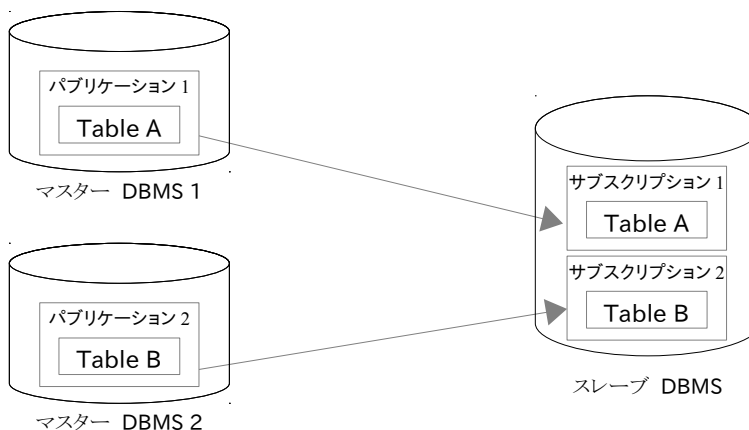
- 1) マスター DBMS 上の全てのテーブルをスレーブ側に複製



- 2) マスター DBMS 上の各テーブルをそれぞれ他ホストの DBMS に分けて複製

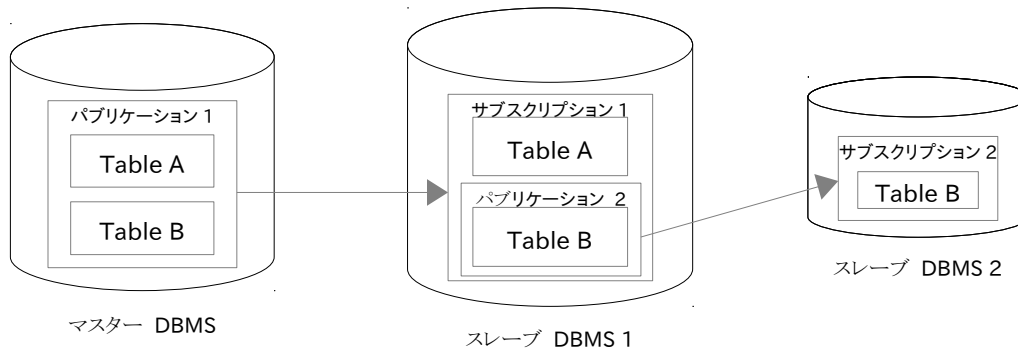


- 3) 複数のホストに点在する DBMS のテーブルを一つのホストの DBMS に統合



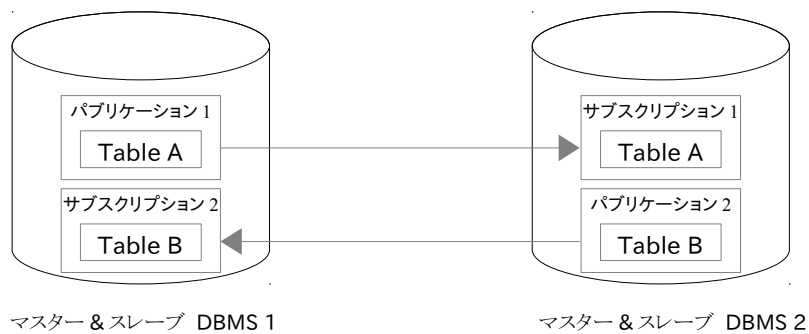
4) カスケードレプリケーション

以下のようにマスター DBMS からサブスクリプション 1 として複製された Table A, B は、他ホストへの複製元のパブリケーションとして、スレーブ DBMS 1 上の Table B のように再び他ホストへ複製する事が可能です。このようにスレーブ DBMS を多段に接続してレプリケーションを行う手法をカスケードレプリケーションと呼称します。



5) 双方向レプリケーション

以下のように各 DBMS 上にあるテーブルを双方向にレプリケーションする事も可能です。このような構成における各データベースは、アプリケーション等からデータ更新を受けるマスター DBMS でもあり、対向ホストからレプリケーションの複製データを受けるスレーブ DBMS でもあります。



5. まとめ

今回は異種 DBMS と PostgreSQL を連携させる商用ソフトウェアの紹介とレプリケーションの設定手順例を中心に、商用ソフトウェアで異種 DBMS を連携させる内容を確認し、作業内容の概要や、注意すべき点について理解を深めました。しかし、実際のエンタープライズ環境では今回挙げた内容の他に、レプリケーションのパフォーマンス要件や、詳細なデータ制御の要件などを十分にテスト・検証して導入する必要があります。

今日においては、異種 DBMS を連携させる商用ツールを用いる事で、各 DBMS の深いスキルなしに異種 DBMS 間で容易かつ柔軟な構成でレプリケーションを実現させる事ができる状況にあります。その為、ユーザは既存の異種 DBMS と相まって動作させる DBMS に PostgreSQL を採用する選択肢があり、ツールを用いて効率的に既存システムに導入する事で、開発や運用コストを抑えつつ効果的なシステム構築をすることができます。

著者

版	所属企業・団体名	部署名	氏名
異種 DB 間連携 第 1 版 (2012 年度 WG2)	サイオステクノロジー株式会社	オープンシステムソリューション部	