

# InfiniDB<sup>®</sup>

## マルチ UM 構成ガイド

Release : 4.5  
Document Version : 4.5-2

InfiniDB マルチ UM 構成ガイド

2014 年 5 月

Copyright © 2014 InfiniDB Corporation. All Rights Reserved.

本書に記載された InfiniDB、InfiniDB ロゴおよびその他のすべての製品またはサービスの名称またはスローガンは、InfiniDB およびそのサプライヤまたはライセンサの商標であり、InfiniDB または当該商標を所有する他社の書面による事前の承諾なしに、全体または一部を複製、模写または使用することを禁じます。

ユーザーは、すべての当該著作権法を順守する責任を負います。著作権に基づく権利を制限することなく、本書のいかなる部分も、InfiniDB の書面による事前の承諾なしに、いかなる形式または手段（電子的、機械的、写真複写的、または記録的手段など）、またはいかなる用途においても、複製、検索システムへの保存または導入、または転送を行うことを禁じます。

InfiniDB は、本書の内容に関して特許（出願中の特許を含む）、商標、著作権、またはその他の知的財産権を保有している場合があります。InfiniDB からの書面によるライセンス契約において明確に許可されている場合を除き、本書の提供により、これらの特許、商標、著作権、またはその他の知的財産権のライセンスが付与されるものではありません。本書の情報は予告なしに変更される場合があります。本書またはその使用による技術的な誤りまたは欠落から生じたいかなる損害に対しても、InfiniDB は責任を負いかねます。

## 目次

1	はじめに.....	4
1.1	対象読者.....	4
1.2	マニュアルリスト.....	4
1.3	マニュアルの入手.....	4
1.4	マニュアルへのフィードバック.....	4
1.5	追加リソース.....	4
2	マルチユーザーモジュールの動作の理解.....	5
2.1	デフォルトのマルチ UM インストール.....	5
2.1.1	問合せ実行.....	5
2.1.2	クロスエンジン結合.....	6
2.1.3	問合せ統計収集.....	7
2.2	マルチフロントエンド構成.....	7
2.2.1	問合せ実行.....	8
2.2.2	クロスエンジン結合.....	8
2.2.3	問合せ統計収集.....	9
3	マルチフロントエンドインタフェースでの同期の保持.....	10
3.1	オプション 1: フロントエンドディレクトリのコピー.....	10
3.2	オプション 2: コマンドの手動実行.....	10
3.2.1	DDL.....	11
3.2.2	InfiniDB 以外の表.....	11
3.3	オプション 3: ユーザーモジュール間のレプリケーション.....	11
4	ローカル PM 問合せ.....	12
4.1	ローカル PM 問合せのインストール.....	13
4.2	UM 間のレプリケーション.....	13
4.3	ローカル PM 設定の構成.....	13
4.4	ローカル PM 問合せの例.....	14
5	付録 A: rsync のスクリプト例.....	15

# 1 はじめに

本書では、InfiniDBユーザーモジュール(UM)コンポーネントのスケールアウト構成について説明します。フロントエンドのパフォーマンスのスケールリングや、フロントエンドの高可用性(HA)のための構成を行う場合に、本書を適用できます。

マルチユーザーモジュール構成のインストールについては、『InfiniDBインストレーションガイド』を参照してください。

## 1.1 対象読者

本書は、InfiniDBシステムのマルチユーザーモジュール構成の管理を行うIT管理者を対象としています。

## 1.2 マニュアルリスト

InfiniDB データベースプラットフォームのマニュアルは、様々な読者を対象とした複数のガイドで構成されています。次の表を参照してください。

マニュアル	説明
『InfiniDB 管理者ガイド』	InfiniDB を管理するための詳細な手順について説明します。
『InfiniDB 概要』	分析用データベース InfiniDB の概要について説明します。
『InfiniDB 最小推奨仕様ガイド』	InfiniDB の実装に必要なハードウェアおよびソフトウェアの最小の推奨仕様を示します。
『InfiniDB インストレーションガイド』	InfiniDB をインストールするために必要な手順の概要について説明します。
『InfiniDB SQL 構文ガイド』	InfiniDB に固有の構文について説明します。
『InfiniDB パフォーマンスチューニングガイド』	分析用データベース InfiniDB をパラレル化および拡張するためのチューニングに役立つ情報について説明します。

## 1.3 マニュアルの入手

英語版のマニュアルは、<http://www.infinidb.co>で入手することができます。追加の支援が必要な場合は[infinidb\\_doc@ashisuto.co.jp](mailto:infinidb_doc@ashisuto.co.jp)にご連絡ください。

## 1.4 マニュアルへのフィードバック

マニュアルの改善に向けて、フィードバック、コメントおよび提案をいただけますようお願いいたします。マニュアル名、バージョンおよびページ番号を添えてコメントを[infinidb\\_doc@ashisuto.co.jp](mailto:infinidb_doc@ashisuto.co.jp)にご送付ください。

## 1.5 追加リソース

InfiniDB のインストールおよびチューニング、または InfiniDB を使用したデータの間合せに関して支援が必要な場合は [infinidb\\_doc@ashisuto.co.jp](mailto:infinidb_doc@ashisuto.co.jp) までご連絡ください。

## 2 マルチユーザーモジュールの動作の理解

InfiniDBでは、複数のユーザーモジュール(UM)を構成できます。このタイプの構成を使用する場合は、あらかじめマルチUMシステムの動作を理解しておく必要があります。

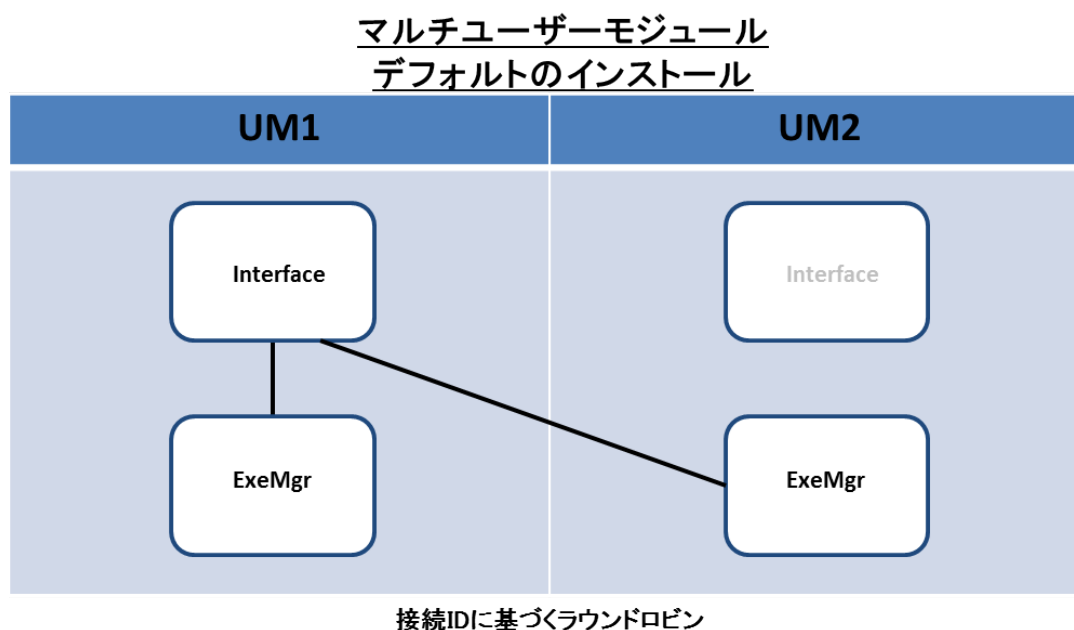
### 2.1 デフォルトのマルチ UM インストール

マルチUMシステムを最初にインストール(初期インストールまたは後からUMを追加)するときには、問合せ実行、クロスエンジン結合および問合せ統計収集について、以下に示す特定の動作があります。

#### 2.1.1 問合せ実行

マルチUMでデフォルトのインストール設定を使用すると、次のメリットがあります。

- ユーザーまたはツールは、UM1のInfiniDBを使用するだけですみます。
- 次の図に示すように、すべてのUM間で問合せのラウンドロビン分散またはスケールアウトが(接続IDに基づいて)自動的行われます。



**注意:**このデフォルト設定では、UM 2からUM  $n$ でmysqlインスタンスが実行されます。これらのmysqlインスタンス間ではスキーマの同期が行われなため、InfiniDBへの接続には使用しないでください。複数のmysqlインスタンスを使用する必要がある場合は、「マルチフロントエンド構成」の項を参照してください。

## 2.1.2 クロスエンジン結合

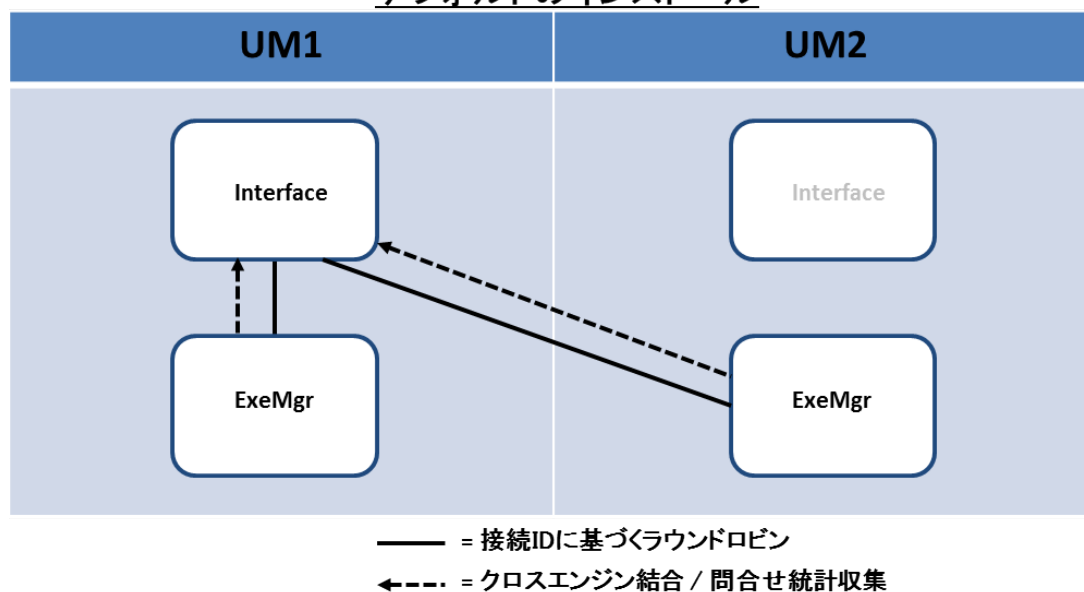
クロスエンジン結合では、問合せ内でInfiniDB表をInfiniDB以外の表(MyISAM表など)と結合することができます。『InfiniDB管理者ガイド』で詳しく説明しているように、InfiniDB以外の表へのアクセスにExeMgrがどのUMを使用するかは、接続情報によって制御します。デフォルトでは、Calpont.XMLファイルのクロスエンジンエントリに接続情報は定義されていません。

```
<CrossEngineSupport>
  <Host>unassigned</Host>
  <Port>3306</Port>
  <User>unassigned</User>
  <Password></Password>
</CrossEngineSupport>
```

この機能を使用する場合は、接続情報を定義し、システムを再起動してCalpont.XMLをすべてのUMに配付することによって、この情報を移入する必要があります。

**注意:**UM1を指すように接続情報を定義すると、どのUMが問合せを処理するかに関係なく、すべてのユーザーがInfiniDB以外の表にアクセスできるようになります。

### マルチユーザーモジュール デフォルトのインストール



UM1 (IPアドレスが10.100.135.85)を指すCalpont.XMLエントリの例:

```
<CrossEngineSupport>
  <Host>10.100.135.85</Host>
  <Port>3306</Port>
  <User>root</User>
  <Password>um_access</Password>
</CrossEngineSupport>
```

#### その他の注意事項:

- 指定したポートに対するファイアウォールアクセスが、UM間で解決される必要があります。詳細はシステム管理者にお問い合わせください。
- アクセスの付与も解決される必要があります。詳細は、『InfiniDB管理者ガイド』を参照してください。

### 2.1.3 問合せ統計収集

これを有効にすると、問合せ統計が収集されて、予約された`infinidb_querystats`データベースの`querystats`表に移入されます。詳細は、『InfiniDB管理者ガイド』の問合せのパフォーマンスに関する項を参照してください。

Calpont.XMLのCrossEngineSupportエントリを使用して、どこにある`querystats`表に移入するかを指定します(前述のクロスエンジン結合の図を参照してください)。問合せおよびクロスエンジン結合と同様に、InfiniDBシステム全体の統計を分析する際にUM1のみを使用すればよいというメリットがあります。

## 2.2 マルチフロントエンド構成

InfiniDBは、同時実行のスケールアウト構成も提供できます。詳細は、『InfiniDB概要』の「InfiniDBのアーキテクチャ」の項を参照してください。

この構成でのメリットは次のとおりです。

- マルチフロントエンドでの高度な同時実行機能があります。
- ユーザーまたはツールは任意のUMのInfiniDBに接続できます。
- UMの1つに障害が発生した場合に備えて高可用性を実現できます。

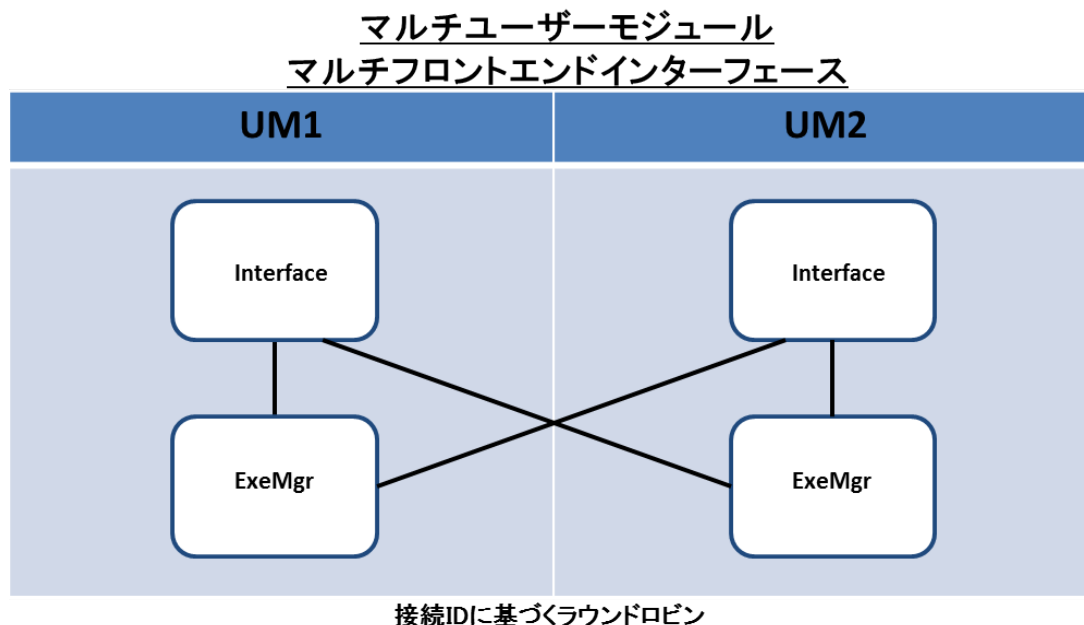
この構成でのデメリットは次のとおりです。

- これらのユーザーモジュールのフロントエンドの定義は、同期がとれている必要があります。そのため、あるUMで実行されたDDLは他のUMにレプリケートする必要があります。
- MySQL構成ファイル(`my.cnf`)を変更する場合は、すべてのUMの同期を手動で保持する必要があります。

**注意:**この構成が初期デフォルト構成と異なるのは、1つのUMだけでなく、すべてのUMのフロントエンドインタフェースを使用するという点だけです。

## 2.2.1 問合せ実行

マルチUMシステムをマルチフロントエンドと併用すると、次の図に示すように、すべてのUMにわたって、問合せ実行で同じ問合せ動作が行われ、同じ自動ラウンドロビン分散が行われます。



ここでも、すべてのUMのフロントエンドの同期を保持することが重要であるということ以外に、この構成を使用するための独自の処理はありません。このマニュアルで後述する「マルチフロントエンドインターフェースでの同期の保持」の項を参照してください。

## 2.2.2 クロスエンジン結合

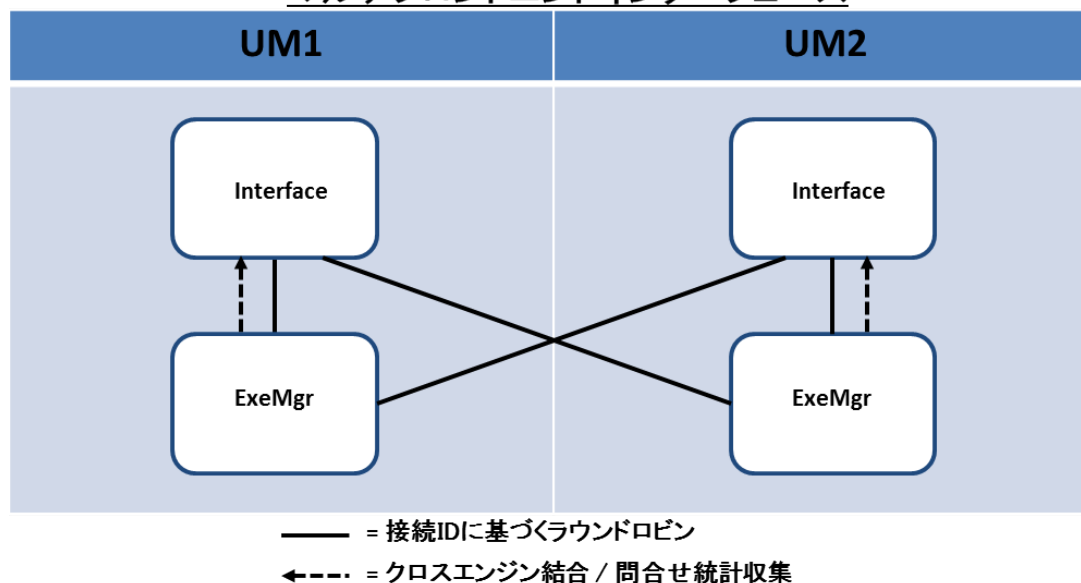
使用しているフロントエンドが単一か複数かに関係なく、クロスエンジン結合を設定する必要があります。この機能を使用する場合は、localhostを指すように接続情報を定義し、システムを再起動してCalpont.XMLをすべてのUMに配付することによって、この情報を移入する必要があります。

注意:

- localhostを指すように接続情報を定義すると、どのUMがクロスエンジン問合せを処理するかに関係なく、すべてのユーザーがInfiniDB以外の表にアクセス可能になり、UMの障害発生時に備えてHAを実現できます。
- クロスエンジン結合に使用されるInfiniDB以外の表はすべて、定義とデータの両方で同期を保持する必要があります。マルチフロントエンドの同期を保持する手順を実行すると、InfiniDB以外の表はInfiniDB構造の下のmysql/dbディレクトリに格納されて自動的に同期がとられます。
- フロントエンドが同期していないと、結果が不正確になる可能性が高くなります。



## マルチユーザーモジュール マルチフロントエンドインターフェース



localhostを指すCalpont.XMLエントリの例:

```
<CrossEngineSupport>
  <Host>127.0.0.1</Host>
  <Port>3306</Port>
  <User>root</User>
  <Password>um_access</Password>
</CrossEngineSupport>
```

### 2.2.3 問合せ統計収集

デフォルトのインストールと同様に、Calpont.XMLのCrossEngineSupportエントリを使用して、どこにあるquerystats表に移入するかを指定します(前述のクロスエンジン結合の図を参照してください)。

問合せ元の各UMで統計が移入され、管理者が統計の分析を各UMで行う必要があることが、マルチフロントエンドインターフェースを使用する際のデメリットになります。

このデメリットが大きい場合は、クロスエンジンと問合せ統計収集の両方で単一のUM(UM1)を指すようにすることをお勧めします。障害が発生した場合は、Calpont.XMLエントリを手動で変更して、別のUM(UM2)を指すようにする必要があります。

### 3 マルチフロントエンドインタフェースでの同期の保持

前述のとおり、マルチフロントエンドインタフェースを使用する場合は、これらのユーザーモジュールのフロントエンド定義の同期を保持する必要があります。そのため、あるUMで実行されたDDLおよびInfiniDB以外の表の作成または移入を、他のUMにレプリケートする必要があります。以下は、レプリケーションを効果的に実行するオプションについて説明しています。

マルチフロントエンドインタフェースがある構成では、すべてのDDLおよびInfiniDB以外の表の作成または移入を、1つのUMのみ(できればUM1)で実行することをお勧めします。こうすると、以下の2つのオプションの1つを使用することにより、すべてのUMで同期を保持しやすくなります。

**注意:**以下のオプションに加えて、MySQL構成ファイル(my.cnf)の変更は、すべてのUMに適用する必要があります。

#### 3.1 オプション 1: フロントエンドディレクトリのコピー

**注意:**このオプションの使用を推奨します。

すべてのフロントエンドスキーマ情報は、InfiniDBシステムの次のディレクトリに存在します。

```
mysql/db
```

このスキーマ情報は、すべてのオンラインUM間で同期を保持する必要があります。同期を保持するには、rsyncコマンドを使用してソースUMを他のすべてのターゲットUMに同期させます。

次に、2つのUMサーバーをrootインストールと同期させるために使用するrsyncコマンドの例を示します(MySQLの.errファイルおよび.pidファイルを除く)。これは、DDL文が実行されるソースUMで実行されます。

```
rsync -vuopg -e ssh --delete --exclude=*err --exclude=*pid -r /
usr/local/Calpont/mysql/db root@infiniDB_UM2:/usr/local/Calpont/ mysql/
```

さらに自動化する必要がある場合は、このコマンドをスクリプトに記述します。スクリプトにこのコマンドを記述する例については、「付録A:rsyncのスクリプト例」を参照してください。

#### 3.2 オプション 2: コマンドの手動実行

次の領域の構文は、すべてのUMで同期を保持する必要があります。

- DDL
- すべてのInfiniDB以外の表のDDL、DMLおよびインポート

### 3.2.1 DDL

CREATEコマンドおよびDROPコマンドは、同期させる必要のあるすべてのUMで使用できます。

- あるUMでCREATE TABLEを実行する場合、「schema sync only」コメントを使用して他のUMと同期させます。他のすべてのUMで、コメントオプションを使用した同じCREATE TABLEコマンドを実行する必要があります。

```
CREATE newtable (column_info) ENGINE=INFINIDB COMMENT='schema sync only';
```

- あるUMでDROP TABLEを実行する場合、RESTRICTオプションを使用して他のUMと同期させます。このDROP TABLEコマンドを、他のすべてのUMで実行する必要があります。

```
DROP TABLE droppedtable RESTRICT;
```

- ALTER TABLEコマンドまたはRENAME TABLEコマンドについては、前述の2つの文(DROPおよびCREATE)を組み合わせて、他のすべてのUMで実行する必要があります。

```
DROP TABLE alteredtable RESTRICT;
CREATE alteredtable (column_info) ENGINE=INFINIDB COMMENT='schema sync only';
```

- CREATE PROCEDUREコマンドまたはDROP PROCEDUREコマンドについては、同じコマンドを他のすべてのUMで実行する必要があります。
- CREATE DATABASEコマンドまたはDROP DATABASEコマンドについては、同じコマンドを他のすべてのUMで実行する必要があります。

### 3.2.2 InfiniDB 以外の表

この手動の方法を使用する場合は、1つのUMで実行したInfiniDB以外の表に対するすべての操作を、すべてのUMで同期させる必要があります。次の操作が対象になります。

- DDL: 表の作成または変更。
- DML: すべてのINSERT、UPDATE、DELETE、LOAD DATA INFILEコマンド。
- インポート: InfiniDB以外の表へのインポートは、すべてのUM間でレプリケートする必要があります。

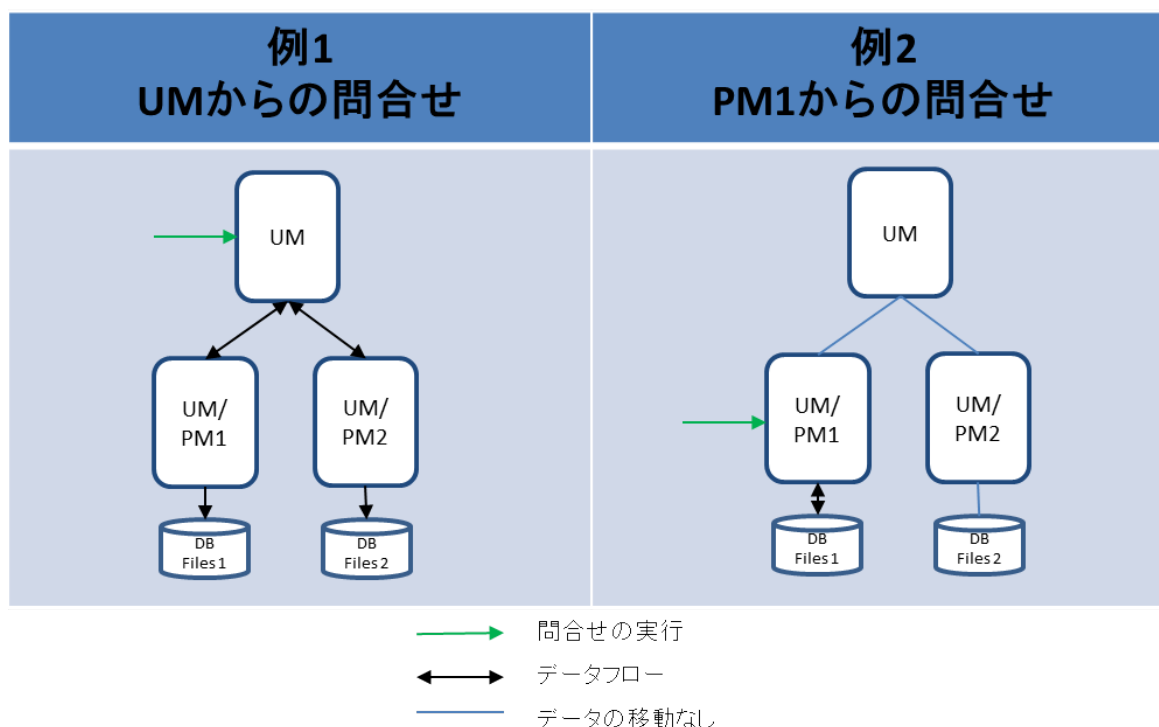
### 3.3 オプション 3: ユーザーモジュール間のレプリケーション

1つのレプリケーションマスターと、1つ以上のレプリケーションスレーブがある場合は、ユーザーモジュール間のレプリケーションを設定できます。詳細は、後述の「UM間のレプリケーション」を参照してください。

## 4 ローカル PM 問合せ

InfiniDB には、単一の PM からデータの問合せを行い、その問合せ結果を PM に残して、UM と PM 間のネットワーク帯域幅を使用しないようにする機能があります。この機能を使用すると、ローカル PM 問合せとその結果を、別の問合せまたは cpimport への入力として使用できます。次の図は、標準的な UM サーバーからの問合せと、PM サーバーからの問合せの違いを示します。

### ローカルPM問合せの比較



例 1 では、UM からの問合せは、すべての PM からデータを取得して問合せを処理する、標準的なパスに従っています。例 2 では、PM からの問合せで、その PM に固有のデータのみを取得しています。

このタイプの処理を利用するには、InfiniDB の次の主要領域で設定を行う必要があります。

- インストール
- フロントエンドデータの同期を保証する UM (スタンドアロンの UM と、ローカル PM を構成する UM の両方) 間のレプリケーション
- ローカル PM 設定の構成

## 4.1 ローカル PM 問合せのインストール

インストール時またはアップグレード時に `postConfigure` ユーティリティを `-lq` オプションを指定して実行することで、InfiniDB の PM にローカル PM 問合せを設定できます。このオプションを使用すると、その PM は UM/PM 複合サーバーに変更されます。ユーザーは、必要に応じて、特定の UM/PM 複合サーバーにログインします。

```
postConfigure -lq
```

**注意:**このオプションでは、すべての UM(スタンドアロンの UM と、PM を構成する UM の両方)でレプリケーションが自動的に有効になることに注意してください。後述の「UM 間のレプリケーション」を参照してください。

## 4.2 UM 間のレプリケーション

前述のとおり、マルチフロントエンドインターフェースを使用する場合は、これらのユーザーモジュールのフロントエンド定義の同期を保持する必要があります。そのため、ある UM で実行された DDL および InfiniDB 以外の表の作成または移入を、他の UM にレプリケートする必要があります。

InfiniDB UM で自動レプリケーションを設定して、1 つの UM(通常は UM1)をレプリケーションマスターに、他のすべての UM(スタンドアロンの UM または UM/PM 複合サーバー)をレプリケーションスレーブにすることができます。これを実行するには、インストール時またはアップグレード時に、`postConfigure` ユーティリティを `-rep` オプションを指定して実行します。

```
postConfigure -rep
```

**注意:**レプリケーションとともにローカル PM 問合せを使用する場合は、`postConfigure -lq` オプションを使用する必要があることに注意してください。「ローカル PM 問合せのインストール」を参照してください。

## 4.3 ローカル PM 設定の構成

InfiniDBには、UMのデータベース全体ではなく、単一のPMのみでデータの問合せを実行する機能があります。これには、`my.cnf`構成ファイルの`infinidb_local_query`変数を使用して、インスタンスのデフォルトとして設定するか、またはセッションレベルで設定します。`postConfigure`ユーティリティを `-lq`オプション(ローカルPM問合せ)を指定して実行していた場合、この変数は各PMに対して自動的に設定されています。この変数を、インスタンスレベルまたはセッションレベルのどちらで変更するか選択できます。インスタンスレベルでの変更については『InfiniDB管理者ガイド』を、セッションレベルでの変更については『InfiniDB SQL構文ガイド』を参照してください。

この変数をオンに設定すると、ローカルPMからのデータのみが返されるようになるので注意してください。

#### 4.4 ローカル PM 問合せの例

##### 例1:ローカルPMの単一の表からのSELECTによるローカルPMへのインポート

infinidb\_local\_query変数を1に設定します(これをローカルPM問合せのデフォルトにします)。

```
idbmysql -e 'select * from source_schema.source_table;' -N |
/usr/local/Calpont/bin/cpimport target_schema target_table -s
'\t' -nl
```

##### 例2:PMノードのファクト表と全ノードのディメンション表の結合を含むSELECTによるローカルPMへのインポート

infinidb\_local\_query変数を0に設定します(これをローカルPM問合せのデフォルトにします)。

次のようなスクリプト(この例ではextract\_query\_script.sql)を作成します。

```
set infinidb_local_query=0;
select fact.column1, dim.column2 from fact join dim using (key)
where idbPm(fact.key) = idbLocalPm();
```

infinidb\_local\_queryが0に設定されて、すべてのPMにわたる問合せが可能になります。

PMノード上のUMプロセスが、(idbLocalPm()関数の使用が示しているように)PMノードからローカルにファクト表データを取得する一方、ディメンション表データはすべてのPMノードから抽出されるように、問合せが構成されます。

次に、これをcpimportに直接送信するスクリプトを実行できます。

```
idbmysql source_schema -N < extract_query_script.sql |
/usr/local/Calpont/bin/cpimport target_schema target_table -s
'\t' -nl
```

## 5 付録 A:rsync のスクリプト例

次に、rsyncコマンドの実行をさらに自動化するためにスクリプトにrsyncコマンドを記述する例を示します。rsyncコマンドは、同期を保持する必要がある各UMにレプリケートされます。

```
#!/usr/bin/expect
#
set timeout 600
set PASSWORD serverpwd
set COMMAND1 "rsync -vuopg -e ssh --delete --exclude=*err --exclude=*pid -r /usr/
local/Calpont/mysql/db root@infinidb_UM2:/usr/local/Calpont/mysql/"
set DEBUG 0
log_user $DEBUG
spawn -noecho /bin/bash
expect -re "# "
#
# send command
#
send "$COMMAND1\n"
expect {
    -re "Host key verification failed" { send_user "FAILED: Host key verification
        failed\n" ; exit -1}
    -re "service not known" { send_user " FAILED: Invalid Host\n" ; exit -1}
    -re "ssh: connect to host" { send_user " FAILED: Invalid Host\n" ; exit -1 }
    -re "authenticity" { send "yes\n"
        expect {
            -re "word: " { send "$PASSWORD\n" } abort
        }
    }
}
-re "word: " { send "$PASSWORD\n" } abort
}
expect {
    -re "# " {} abort
    -re "Permission denied" { send_user " FAILED: Invalid password\n" ; exit -1 }
    -re "(y or n)" { send "y\n"
        expect -re "# " { exit }
    }
}
}
exit
```