

## FioranoMQ のクラスタリング

### メッセージング プラットフォームの障害回復と負荷分散



**Fiorano**  
Enabling change at the speed of thought

Fiorano Software, Inc.  
日本オフィス  
東京都千代田区外神田 3-13-2  
03-6777-7530  
メール: [info\\_jp@fiorano.com](mailto:info_jp@fiorano.com)  
web サイト: [www.fiorano.com/jp/](http://www.fiorano.com/jp/)

Entire contents © 2006 Fiorano Software, Inc. All rights reserved.

この文書は、書面による事前の許可なくいかなる形態においても複製を作成することを禁止されています。この文書に記載されている情報は、信頼がおけると信じるに足る供給元から得ています。

Fiorano Software 社は、この文書の内容について、正確性および完全性の保障をするものではありません。Fiorano Software 社は、この文書に記載されている情報およびその翻訳物の誤記、脱落または不十分性について責任を負いません。

Fiorano Software 社は、予告なくこの文書に記載されている内容および意見を変更することがあります。

## ▶ はじめに

実業で使われているメッセージング アプリケーションでは、多量 / 大容量のメッセージが交換され、たいへん負荷の大きなものとなっています。このような状況では、市販されている高性能なハードウェアにメッセージング サーバーをインストールしても、十分に満足いくものとはなりません。この問題を解決するのにより適した (費用面においてもスケーラビリティ特性の面においても) ソリューションがあるはずですが、最も適したソリューションは、“n” 個のメッセージング サーバーを相互に接続し、これらのサーバーの間で付加を分散させることです。このような “n” 個のサーバーの論理的な固まりを **クラスター** と呼びます。クラスターはまた、単一のサーバーでは不可能なフェイルオーバーをサポートすることができます。

本ホワイトペーパーは、FioranoMQ をメッセージング プラットフォームとして利用した場合の堅牢性、スケーラビリティの高さを理解していただくことを目的に、FioranoMQ が備えているクラスタリング機能を紹介します。

## ▶ よくある問題

サーバーが一時的に接続要求を受けられない状況になることがあります。例えば、大多数のクライアントが同時接続している状態では、socket バッファの空きがなくなってしまうことがあります。この問題の解決に最も適したクライアント側の対処方法は、再接続を試みることです。バッファの不足は一時的なものであり、再試行の間にバッファが解放される可能性があります。

一時的なオーバーロード状態になるのではなく、サーバーが完全にダウンしてしまう場合もあります。この場合の対処方法は、クライアントがバックアップ サーバーに接続できるようにしておくことです。バックアップ サーバーのリストを接続ファクトリ オブジェクトのパラメータとして扱うようにしてしまうと、クライアントのプログラムコードのポータビリティが失われてしまいます。

この章では、このようなよくある問題に対する望ましい対処方法について説明し、次の章では FioranoMQ がどのような対処機能を備えているかを説明します。

### サーバー接続のロスト

この問題の発生率は比較的高く、必ず対処できるようにしておかなければなりません。クライアント サイドの永続化も必須となります。ストア & フォワード機能のサポートによってクライアントが非接続モードで動作できるようになり、メッセージの損失を避けられるようになります。クライアント サイドの永続化機能を実現する場合には、トランザクション セッションのサポートと、メッセージの二重化に対する対処が必要となります。

### サーバー リソースの不足

次にリストアップしたようなサーバー リソースが不足状態になることで、サーバーへの接続ができなくなることがあります。

- 接続
- RAM やディスク容量などのハードウェア リソース
- スレッド
- ファイル ディスクリプタ

## ソケット その他

サーバーをクラスター化することで、リソース量を増やすことができます。クラスター化によって、接続要求をその時点で負荷の軽いサーバーに振り分けたり (**ロード バランシング**)、プライマリ サーバーに接続できなくなった場合にスタンバイ サーバーが替わりに立ち上がったり (**フェイルオーバー**) することで、リソース不足に対処します。クライアントプログラムのポータビリティを確保するためには、クライアントにはクラスターが1つのサーバとして見える (トランスペアレントである) 必要があります。ロードバランシングやフェイルオーバーは、クライアントにとってトランスペアレントになっていて、さらにこの両方を同時に組み合わせて使用できる必要もあります。リソース不足が一時的なものである可能性もあるため、クライアント側では代替サーバーへ接続する前に再接続を試みるのも良い方法です。

### **フェイル オーバー**

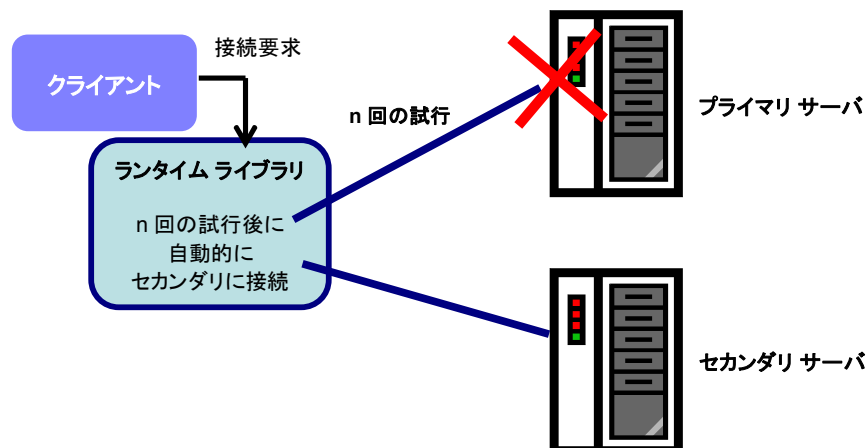
1つのサーバーがクラッシュした場合、そのサーバに接続していたクライアントは、セカンダリ サーバーに接続することで処理を継続できるようにしておく必要があります。これは、“**フェイルオーバー**” と呼ばれる方法です。サーバーが復帰した際には、セカンダリ サーバーから処理を引き継ぎます。“**ホット フェイルオーバー**” とは、シームレスに処理を引き継ぎ、遅延をほぼ発生させない方法です。このためには、セカンダリ サーバーはアップ状態とし、常に処理ステートとメッセージ データにアクセスできる状態になっている必要があります。

## ▶ FioranoMQ 2006 のソリューション

上述の問題を解決するために、FioranoMQ には以下の機能が備わっています。

### クライアントからの接続時の自動フェイルオーバー

1つのコンポーネントの障害によってシステム全体がダウンしてしまうことは避けなければなりません。そのため重要なコンポーネントのバックアップは、高可用性を実現するために不可欠なものとなります。FioranoMQ 2006 では、サーバーへの接続を試みているクライアントのバックアップサーバーへのフェイルオーバーが自動化されています。あるクライアントアプリケーションがサーバーに接続を試みている場合、接続をN回(設定された回数) 試行します。それでも接続に失敗した場合には、ランタイムライブラリがバックアップサーバーへの接続を行います。バックアップサーバーの指定は、あらかじめ接続ファクトリにそのURLを設定しておきます。すなわち、この処理は、Fioranoのランタイムインボケーションによって自動的に行われ、クライアント内でバックアップサーバーのリストを検索するようなプログラミングを行う必要はありません。これは、ベンダーのロックイン(囲い込み)を防ぎ、クライアントプログラムをJMSに完全に準拠したプログラムとしてコーディングできることを意味します。



### トランスペアレンシーとプログラムコードのポータビリティ

フェイルオーバー自動化のメカニズムは、クライアントにとって完全にトランスペアレントである必要があります。FioranoMQ では、再接続のメカニズムをクライアントから隠すことによって、このトランスペアレンシーを確保しています。再接続はランタイムライブラリによって自動的に行われます。この自動化に必要な作業はすべてサーバーのコンフィグ設定のみであり、クライアントプログラムの作成に特別なAPIを必要しないため、JMSに準拠したクライアントプログラムを作成でき、FioranoMQ以外のJMSサーバーとも連携させることができます。

### コンフィグレーションパラメータ

FioranoMQは、高度にデザインされたコンフィグレーションパラメータを備えており、サーバーの機能をユーザーの環境に合わせてカスタマイズできます。このパラメータにはクライアントアプリケーションの振る舞いを設定するものが多く含まれています。こうすることによって、クライアントプログラムのトランスペアレンシーを高めています。

## 管理システム (Admin System)

管理オブジェクト (接続ファクトリ、キュー、トピックなど) を管理するための管理システム (Admin System) が、サーバーにまず最初に必要となるものです。管理システムはクライアント毎に異なります。また、そのバックアップも必要となります。クライアントにとっては、管理システムのプライマリおよびセカンダリの設定はトランスペアレントになっていなければなりません。FioranoMQ では、管理システムの格納方法を任意に設定できるようになっています。

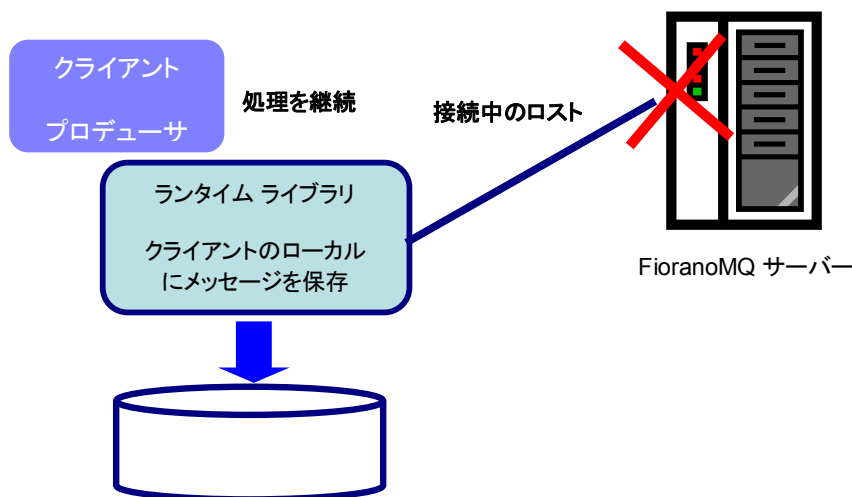
- JNDI 準拠のディレクトリ サービス
- RDBMS サーバー
- XML ファイル
- Fiorano 独自のファイル フォーマット (JNDI を用います)

上記のものに加え、ユーザー独自の管理システムを作成することも可能です。

## 接続中のロスト

### サーバーへの接続のロスト (クライアントサイドでのメッセージの永続化)

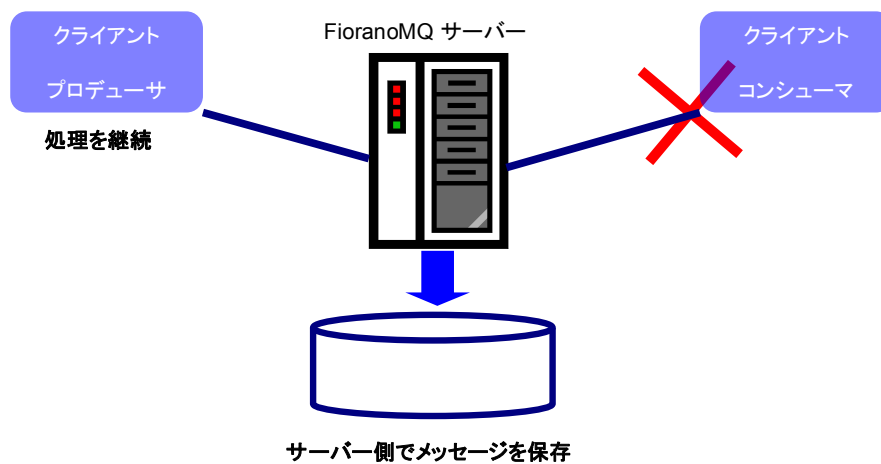
前述したように FioranoMQ では、クライアントに自動フェイル オーバーの機能を提供しています。このメカニズムは、既存の接続に障害が発生した場合と、新規の接続を生成しようとしている場合の両方に有効です。バックアップ サーバーを設定していない状態で既存の接続がロストした場合でも、クライアントはメッセージの送信処理を継続することができます。このような状況で送信されたメッセージは、クライアントが稼動しているマシンのローカル リポジトリに保存されます。その間、ランタイム ライブラリはサーバーへの再接続を試みます。再接続に成功した時点で、ローカル リポジトリに保存されていたメッセージが自動的に送信されます。



### サーバーからクライアントへの接続のロスト (恒久接続)

メッセージ コンシューマ (レシーバー) が恒久接続を設定している状態でコンシューマとの間の接続がロストした場合、サーバーはプロデューサー (センドー) から送られてきたメッセージを保存します。このメッセージは、コンシューマが次に接続したときに受け取ることができます。一方、コンシューマのランタイム ライブラリは、一定時間毎に接続に対するピングを行っています。この機能によって、JVM がネットワーク障害を検出するよりも早い

タイミングで JMS サーバーが障害を検出できます (JVM が障害を検出するタイミングは、ソケットに次の書き込み処理を行った時点です。メッセージのコンシューマは書き込み処理を通常行わないため、JVM がネットワーク障害を検出するタイミングは大幅に遅れてしまいます)。ランタイムライブラリのピング機能によって、コンシューマはいち早く死んだソケットの検出を行いクリーンアップ処理を行うことができ、サーバーとの再接続処理を開始できます。



## 負荷分散とスケーラビリティ

### サーバーのリソース不足

この問題に対処するために、FioranoMQ 2006 では“**ディスパッチャ**”と呼ばれるロードバランシングを行うためのコンポーネントを用意しています。

詳細については、後述します。

### サーバー間のコミュニケーション

複数のメッセージングサーバーにまたがった情報交換は、現実の世界ではよくあることです。FioranoMQ では、“**リピータ**”および“**ブリッジ**”と呼んでいるコンポーネントを用いて、複数サーバーにまたがった情報交換を実現しています。リピータは、fioranoMQ 間の通信に、ブリッジは FioranoMQ と他社製品 (IBM WebspherMQ、MSMQ、Tibco Rendezvous など) との間の通信に用います。

詳細については、後述します。

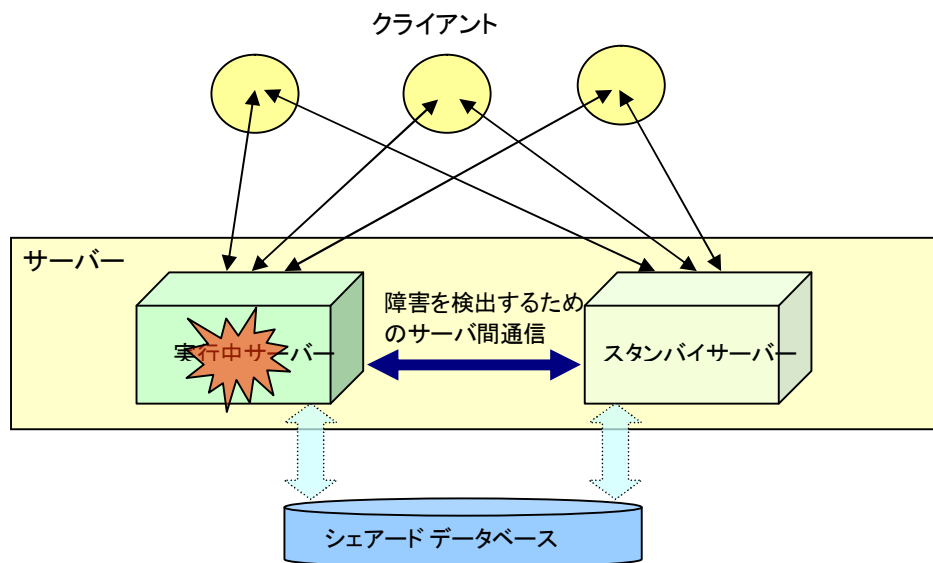
### スケーラビリティ

ロードバランシングや自動フェイルオーバーのメカニズムは、JMS サービスを同時に利用するクライアント数において無限のスケーラビリティを提供するもので、単一のクラスタ構成で数千のクライアントの同時接続をサポートできます。サーバー間通信と組み合わせることで、Fiorano のクラスタ構成はより堅牢でスケーラビリティの高いものとなります。

なお、FioranoMQ には、SCM (スケーラブル コネクション マネージャ) と呼ばれる 単一のサーバー上で大多数な同時接続をサポートするためのコンポーネントも用意されています。これについては、弊社の製品ドキュメントを参照してください。

#### 高可用性 (HA : High Availability) 機能

FioranoMQ には、上述した障害回復の手段のほかに、高可用性 (HA : High Availability) 機能も備えています。メッセージの永続化や恒久接続では、障害発生時にメッセージを一時的に保存しておき、サーバーやコンシューマクライアントが障害から回復した時点でメッセージを送信する機能です。HA では障害発生時にスタンバイサーバーが自動的に立ち上がり、プライマリサーバーの状態をそのまま引き継いで処理を続行します。このため、メッセージが遅延なく送信されます。



上図では、シェアードデータベース方式を採っていますが、それぞれのサーバーに専有にデータベースを設け、データベースのミラーリングを行う方法も採れます。

FioranoMQ の HA 機能は、すべてソフトウェア (FioranoMQ サーバー) で実現されており、HA のための専用のハードウェア装置を必要としません。

HA の詳細に関しては、製品ドキュメントを参照してください。

## ▶ クラスターリング コンポーネント

FioranoMQ のクラスターリング機能を強化するコンポーネントに、

- ディスパッチャー
- リピーター
- ブリッジ

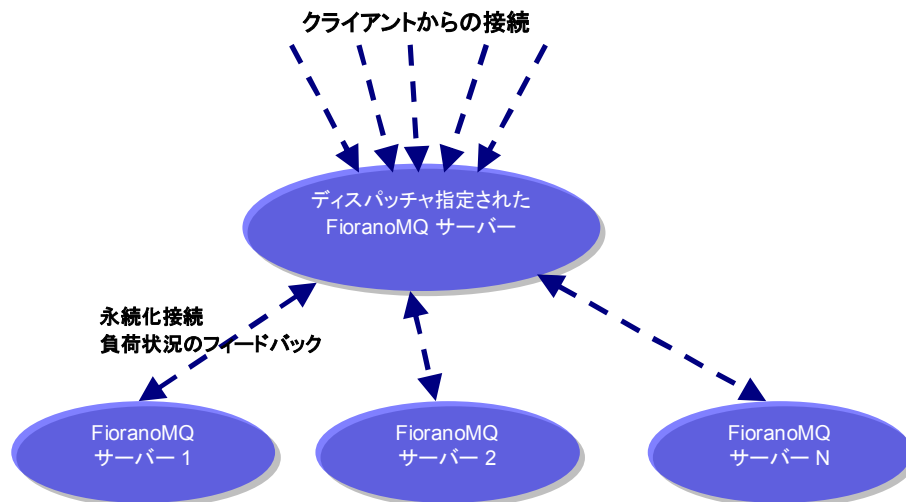
の 3種類があります。

**ディスパッチャー**は、クライアント接続のロードバランシング (負荷分散) に用いられるもので、クラスター内のサーバー間で負荷を分散します。

**リピーター**と**ブリッジ**は、複数のサーバーにまたがったトピックやキューサーバーのメッセージングに用いられるものです。FioranoMQ 同士のメッセージングにとどまらず、他社の JMS 製品や MQ 製品 (非 JMS) との間のメッセージングをサポートします。

### FioranoMQ ディスパッチャー

FioranoMQ のロードバランシング (負荷分散) では、下図に示すようにディスパッチャーがクラスター内で最も負荷の少ないサーバーにクライアントからの接続要求を渡すことで実現しています。'



ディスパッチャは、ディスパッチャとして稼動することを指定された FioranoMQ サーバーです。サーバーのコンフィグ設定で簡単にそのオン、オフが行えます。ディスパッチャ機能がオンに設定されていると、サーバーはクライアントからの接続をそのクラスターメンバーに渡します。クライアントのプログラムにおいては、この機能を利用するために特別な API やコーディングを必要としません。

クラスターにメンバーを追加したり削除することは、いつでも自由に行えます。クラスター内のメンバーが何らかの理由によってダウンした場合には、そのサーバーに接続されているクライアントはすべて別のサーバーに渡され



ます。また、ディスパッチャでは自動フェイルオーバー機能が働き、ディスパッチャがダウンした場合には指定されたセカンダリのディスパッチャに引き継がれます。セカンダリ ディスパッチャの指定は、クラスタ内のメンバーから選択して指定します。

トピック接続ファクトリ、トピック、キュー接続ファクトリ、キュー、ユニファイド接続ファクトリは、クラスタ内のすべてのサーバー (ディスパッチャも含む) に存在し、クライアントからの新たな接続があるたびにすべてのサーバーで同じように作成されます。つまり、すべてのサーバーで同じものを複製して持つことになります。こうすることで、いずれかのサーバーがダウンした場合に、クラスタ内の他のサーバーにクライアント接続を引継ぎ、処理を継続するフェイルオーバーが実現できます。

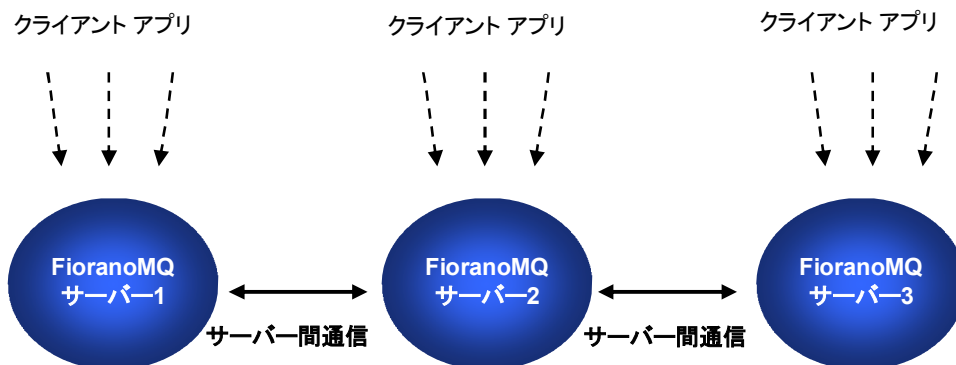
ディスパッチャとクラスタ内のメンバーとの間に、永続化接続 (Persistent connection) が確立され、メンバーからディスパッチャへ負荷情報 (現在接続しているクライアント数など) を渡す経路となります。この情報によって、ディスパッチャはメンバーの負荷状況を知ることができます。

## FioranoMQ リピーター

FioranoMQ では、複数の FioranoMQ サーバーを繋ぐことができ、あるサーバーに接続しているクライアントと別のサーバーに接続されているクライアントとの間で通信することができます。この機能は、地理的に離れたアプリケーション同士の情報交換に有効です。

下の図において、サーバー間通信を担っているのが、FioranoMQ リピーターです。リピーターによる接続は、LAN 経由でも WAN 経由でもかまいません (TCP/IP、SSL/HTTP/HTTPS)。

また、リクエスト-リプライも可能となっています。



FioranoMQ の管理ツールによって、サーバー間のメッセージの転送経路を自由に設定することができます (例えば S1 から S2 または S1 から S3 へ送信)。ネットワーク障害が発生した場合でも、クライアントは影響を受けません。プロデューサークライアントはメッセージの送信を続けられ、送信されたメッセージはローカルのサーバーに保存されます。サーバーが障害から回復した時点で、保存されたメッセージが送信されます。

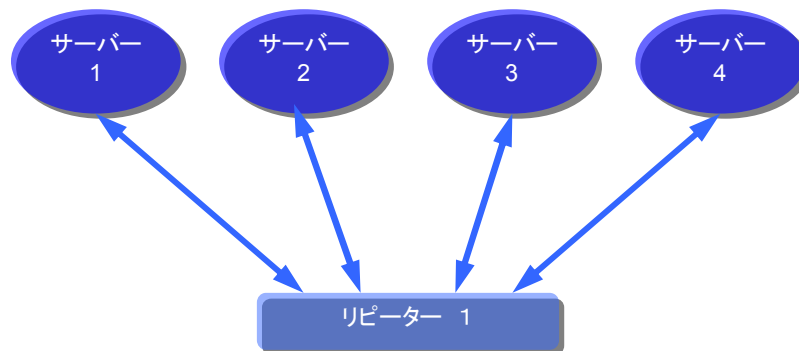
### サーバー間通信のメカニズム

FioranoMQ サーバー間の通信を担っているのが、FioranoMQ リピーターです。FioranoMQ リピーターは、異なるサーバーに接続されているクライアントへメッセージを配信する役割を果たす、専用の JMS クライアントです。

サーバーの接続の形態は自由であり、次のようなトポロジーが考えられます。いずれのトポロジーにおいても、FioranoMQ サーバーは、LAN もしくは WAN 上のどちらに存在していてもかまいません。

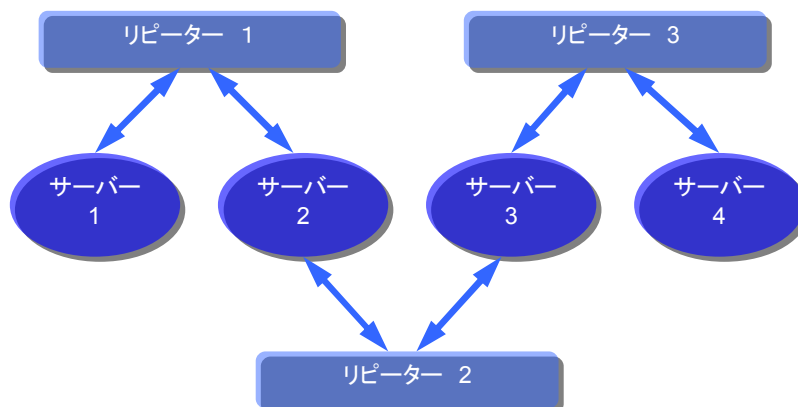
### ハブ & スポーク

次の図は、ハブ & スポークを基本とした複数の FioranoMQ サーバーの配置を示しています。FioranoMQ リピーターは、すべてのサーバー間のメッセージ送受信を行います。したがって、例えばサーバー 1 のクライアントからサーバー 2 もしくは サーバー 3 またはサーバー 4 のクライアントへメッセージを渡すことが可能となります。



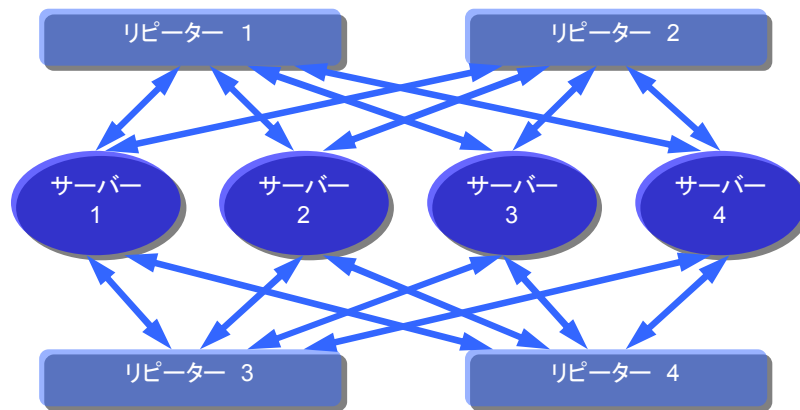
### バス

下の図は、バス形式のトポロジーに基づくものです。この例では、リピーター 1 は、サーバー 1 のクライアントとサーバー 2 のクライアントとの間のメッセージしか送受信できず、サーバー 3 のクライアントとのメッセージ送受信は行えないことに注意してください。



### メッシュ

メッシュトポロジーにおいては、すべてのサーバーとつながったリピーターを複数配置します。これは、上述のハブ & スポークと同じサーバー間通信となりますが、リピーターが複数あることでリピーターの負荷分散が図れます。



### FioranoMQ ブリッジ

FioranoMQ ブリッジは、他社のメッセージング サーバー製品との間のインターオペラビリティを実現するためのコンポーネントです。JMS に準拠したサーバー製品との通信は、JMS API を用いて行うことができます。MSMQ との間の通信には、MSMQ Java API を用いる必要があります。同様に、Tibrv との間では、Tibrv Java API を用いる必要があります。FioranoMQ ブリッジは、このような API の違いを吸収します。これによって、FioranoMQ のクライアントは、他のメッセージング サーバー (JMS サーバー、非 JMS サーバーを問いません) のクライアントとの間でメッセージの送受信が行えるようになります。



### ソースキュー (SQ) とターゲットキュー (TQ)

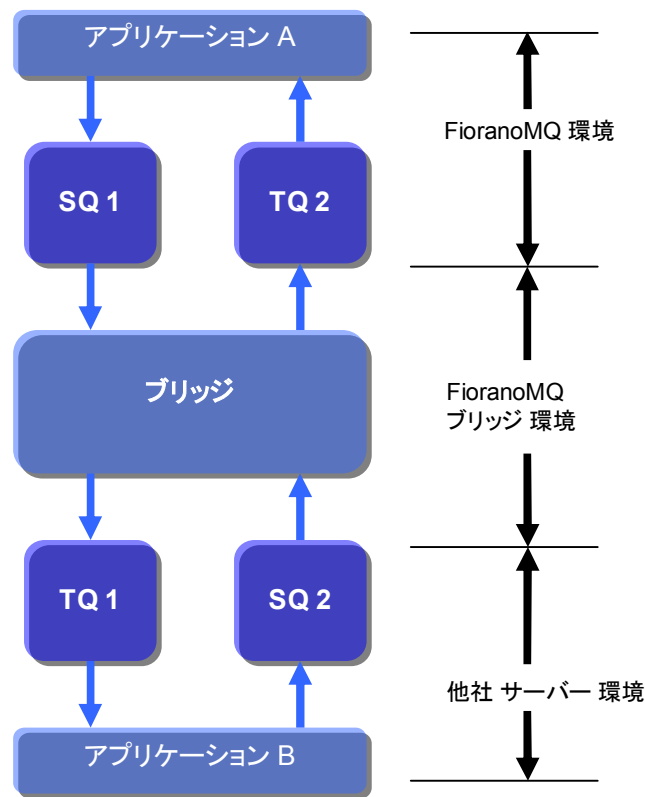
ブリッジによって FioranoMQ と他社のメッセージング サーバーとの間でメッセージ送信を行う際には、ソース キュー (SQ) およびターゲット キュー (TQ) と呼ぶブリッジ用の 2つのキューを用意します。このキューは、それぞれのサーバーでコンフィグします。

- **ソースキュー (SQ):** クライアントアプリケーションは、常にソース キュー (SQ) に対してメッセージを送信します。SQ に送られてきたメッセージは、ブリッジによって読み取られ、対応するターゲット キュー (TQ) に転送されます。SQ は、FioranoMQ のキューとしても、あるいは相手側の他社メッセージング サーバーのキュー

一としても存在します。

- **ターゲットキュー (TQ)**: ターゲットキュー (TQ) は、メッセージ送信の最後に位置するキューです。ブリッジは SQ から受け取ったメッセージを対応する TQ に転送し、相手側クライアントアプリケーションがメッセージを受け取れるようにします。TQ は、FioranoMQ のキューとしても、あるいは相手側の他社メッセージングサーバーのキューとしても存在します。

次の図は、SQ、TQ、ブリッジの関係を示しています。例えば、アプリケーション A は ATM の処理を行うアプリケーションで、メッセージ交換に FioranoMQ を使用しているものとします。このアプリケーションは、口座情報の問い合わせリクエストを受け取るとアプリケーション B にそのリクエストを転送し、アプリケーション B からリターンされてきた口座情報を受け取るよう設計されています。アプリケーション B は、口座情報を検索するアプリケーションで、メッセージ交換に IBM の WebSphereMQ を用いています。



1. アプリケーション A が、口座情報検索のメッセージを SQ1 に送ります。この SQ 1 は、FioranoMQ のキューです。
2. ブリッジが SQ1 上のメッセージを読み取ります。次に、メッセージヘッダーのデータを IBM WebSphere MQ のフォーマットに変換します。そして、TQ1 に変換後のメッセージを送ります。この TQ 1 は、WebSphere MQ のキューです。
3. アプリケーション B は、TQ 1 からメッセージ (口座情報検索リクエスト) を受け取ります。

検索された口座情報は、上の手順と同様に SQ2 → ブリッジ (ヘッダーの変換) → TQ2 と流れ、アプリケーション A によって受信されます。

## ▶ Fiorano Software について

Fiorano Software は、カリフォルニアに本社を置く、エンタープライズ インテグレーション ミドルウェアの業界をリードしている企業で、メッセージング インフラストラクチャ技術において数多くのお客様から高い信頼をよせられています。Fiorano のソリューションは、インターオペラビリティ、パフォーマンス、スケーラビリティ、ROI などの面で新たなパラダイムをもたらしています。アメリカン エクスプレス、AT&T ワイヤレス、ボーイング、BP (旧ブリティッシュ ペトロリアム)、エリクソン、FedEx、ロッキード マーチン、モーガン スタンレイ、モトローラ、POSCO、シュルンベルグなどの世界的なリーダー企業で Fiorano の技術が採用され、企業のバックボーンシステムとして稼動しております。

Fiorano Software に関する詳細な情報は、弊社のホームページ ([www.fiorano.com/jp](http://www.fiorano.com/jp)) をご参照くださるか、[mailto:info\\_jp@fiorano.com](mailto:info_jp@fiorano.com) 宛てに電子メールでお問い合わせください。