

ストリーミングレプリケーション

仕組み、構成のポイント

技術を知る



<input type="checkbox"/> 導入／環境設定	<input type="checkbox"/> 移行	<input type="checkbox"/> 性能	<input type="checkbox"/> チューニング	<input type="checkbox"/> バックアップ／リカバリー
<input checked="" type="checkbox"/> 冗長化／負荷分散	<input type="checkbox"/> 監視	<input type="checkbox"/> データ連携	<input type="checkbox"/> 災害対策	<input type="checkbox"/> 豆知識

データベースでの「レプリケーション」とは、データベースのレプリカ（複製）を作成することです。レプリケーションにより、障害が発生しても継続して稼働でき、可用性の高いシステムを構築することができます。また、複製されたデータベースを活用し、参照系の SQL を処理させることで、システム全体で、より多くの処理をすることができます。遠隔地にデータベースを複製することで、災害対策にも利用できます。

PostgreSQL のレプリケーションには、データベースクラスタを一括して複製する「ストリーミングレプリケーション（物理レプリケーション）」と、テーブル単位やデータベース単位に複製する「ロジカルレプリケーション（論理レプリケーション）」などがあります。

ここでは、データベースクラスタを一括して複製する「ストリーミングレプリケーション」の仕組みと構成のポイントについて説明します。

なお、本資料では、PostgreSQL 11 で設定可能なパラメーターを使用して説明しています。各パラメーターのデフォルト値は、PostgreSQL 11 でのデフォルト値を記載しています。

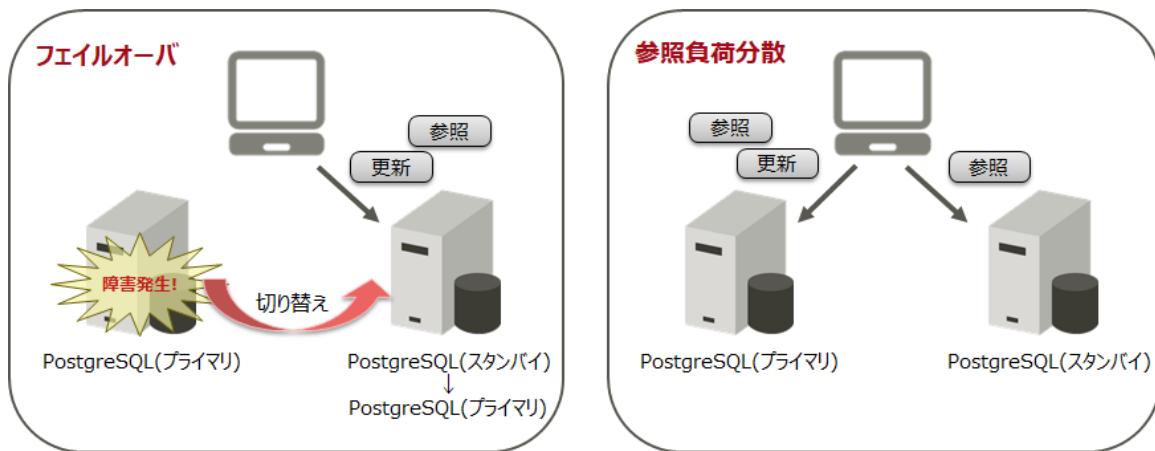
ポイント

PostgreSQL 12 では、設定ファイルの `recovery.conf` に指定するパラメーターが、`postgresql.conf` に統合されました。PostgreSQL 12 を使用する場合、`recovery.conf` が存在すると PostgreSQL サーバーが起動しないなどの仕様変更があります。本資料の `recovery.conf` に関する記事は、別途仕様を確認してください。仕様の詳細については、"PostgreSQL Documentation" の "Release Notes" の "Migration to Version 12" を参照してください。

1. ストリーミングレプリケーションとは

PostgreSQL の標準機能であるストリーミングレプリケーションは、プライマリーサーバーの更新情報を、リアルタイムでスタンバイサーバーに転送することで、プライマリーサーバーとスタンバイサーバーのデータベースを同じ状態に保つことができます。ストリーミングレプリケーションを利用することで、以下のような運用を行うことができます。

フェイルオーバー	プライマリーサーバーに障害が発生してもスタンバイサーバーで運用を引き継ぐことができます。
参照負荷分散	参照系の SQL 処理を複数のサーバーに分散できます。



プライマリサーバ: 更新および参照ができる
スタンバイサーバ: 参照のみできる

参考

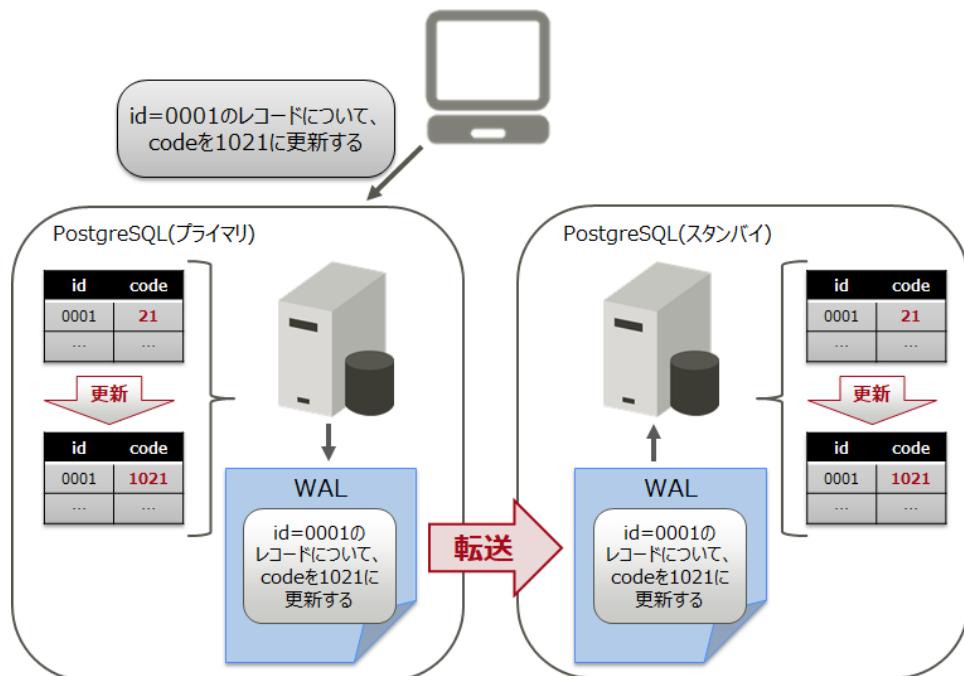
スタンバイサーバで参照系の SQL 处理を実行することで、参照処理の負荷を分散させることができます。ただし、PostgreSQL にクエリやサーバー負荷を考慮した振り分け機能は存在しません。周辺 OSS である Pgpool-II の「負荷分散機能」を利用することによって、参照クエリを各 PostgreSQL に効率的に振り分けられるようになり、負荷が軽減できます。

2. ストリーミングレプリケーションの仕組み

ストリーミングレプリケーションの仕組みを理解するために、何を転送するのか、どのように転送するのかを、詳しく説明します。

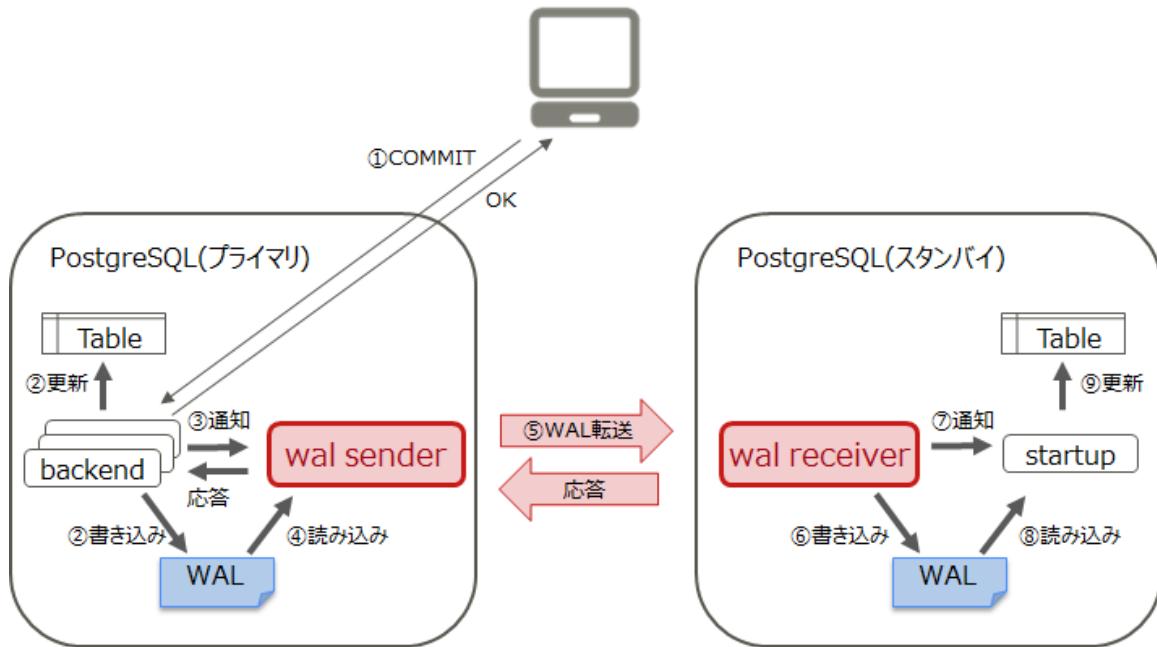
2.1 何を転送するのか (WAL: Write-Ahead Log)

PostgreSQL は、クラッシュリカバリーやロールバックに備え、プライマリーサーバーの更新情報をトランザクションログ（以降、WAL と略します）として保存します。ストリーミングレプリケーションは、リアルタイムにこの WAL をスタンバイサーバーへ転送し、スタンバイサーバーでその WAL を適用する仕組みで動いています。



2.2 どのように転送するのか (wal sender / wal receiver プロセス)

プライマリーサーバーとスタンバイサーバーでの WAL のやりとりは、プライマリーサーバー側の「wal sender プロセス」と、スタンバイサーバー側の「wal receiver プロセス」で行います。postgresql.conf、pg_hba.conf、recovery.conf のパラメーターを設定することにより、これらのプロセスが起動されます。プライマリーサーバー / スタンバイサーバーで、設定するファイルやパラメーターが異なるので、注意してください。設定するパラメーターについては、本ページ内の「【付録】」を参照してください。

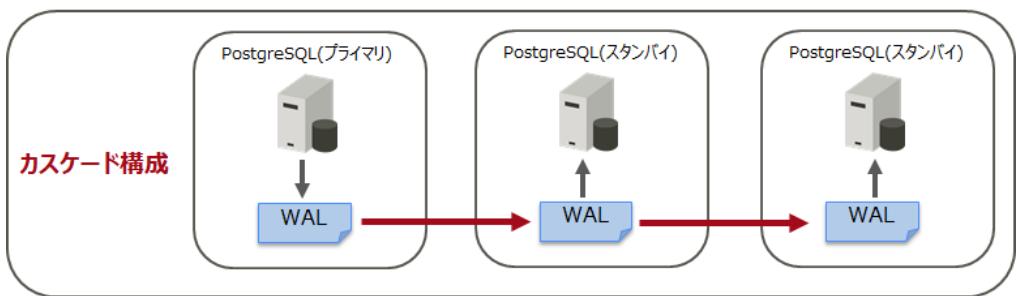
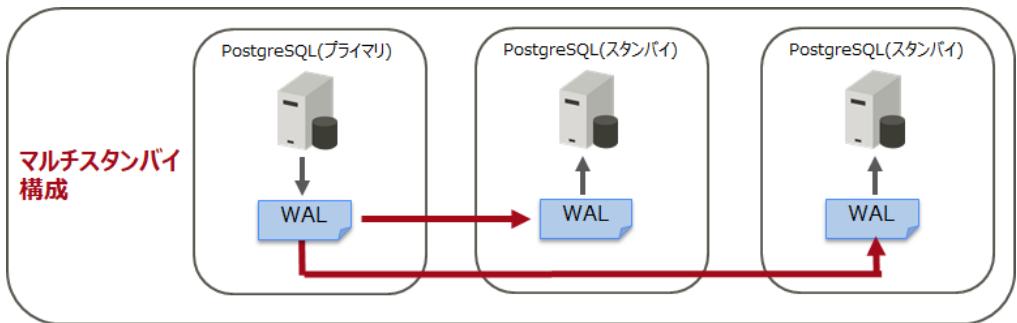


3. 構成

ストリーミングレプリケーションで構築可能な構成、および同期レプリケーションと非同期レプリケーションについて説明します。

3.1 「マルチスタンバイ構成」と「カスケード構成」

ストリーミングレプリケーションは、「1 対 N」の構成で構築することができます。プライマリーサーバーは 1 台のみですが、スタンバイサーバーは複数台用意することができます。プライマリーサーバーから全スタンバイサーバーにつなぐ（WAL を転送する）構成を、「マルチスタンバイ構成」と呼びます。スタンバイサーバーにスタンバイサーバーをつなぐ（WAL を転送する）「カスケード構成」も構築することができます。



3.2 「同期レプリケーション」と「非同期レプリケーション」

ストリーミングレプリケーションは、スタンバイサーバーごとに「同期レプリケーション」とするのか、「非同期レプリケーション」とするのか、選択することができます。ここでは、同期レプリケーションと非同期レプリケーションの特徴と、設定時のポイントについて説明します。

3.2.1 同期レプリケーションと非同期レプリケーションの特徴について

同期レプリケーションと非同期レプリケーションの違いは、スタンバイサーバーからの応答を待ってからプライマリサーバーの処理を完了するかしないかです。以下にそれぞれの特徴を示します。SQL処理のレスポンス時間や高可用性の確保に影響するため、運用にあわせて選択してください。

- **同期レプリケーション**

- スタンバイサーバーからの応答を待ってからプライマリサーバーは処理を完了します。そのため、レスポンス時間に転送時間が含まれます。
- スタンバイサーバーへの WAL 転送の遅延がないため、スタンバイサーバーのデータの最新性（信頼性）が向上します。
- フェイルオーバーおよび参照負荷分散の運用に向いています。

- **非同期レプリケーション（デフォルト）**

- スタンバイサーバーからの応答を待たずにプライマリサーバーの処理を完了します。そのため、ストリーミングレプリケーションを利用しない場合と同等程度のレスポンスです。
- スタンバイサーバーへの WAL 転送および適用（データ更新）が非同期で行われるため、プライマリサーバーで更新された結果がスタンバイサーバーで即座に参照できない（データの状態が古い）場合があります。そのため、フェイルオーバーのタイミング次第で、データが損失する可能性があります。
- 遠隔地へのレプリケーション（災害対策システム）に向いています。

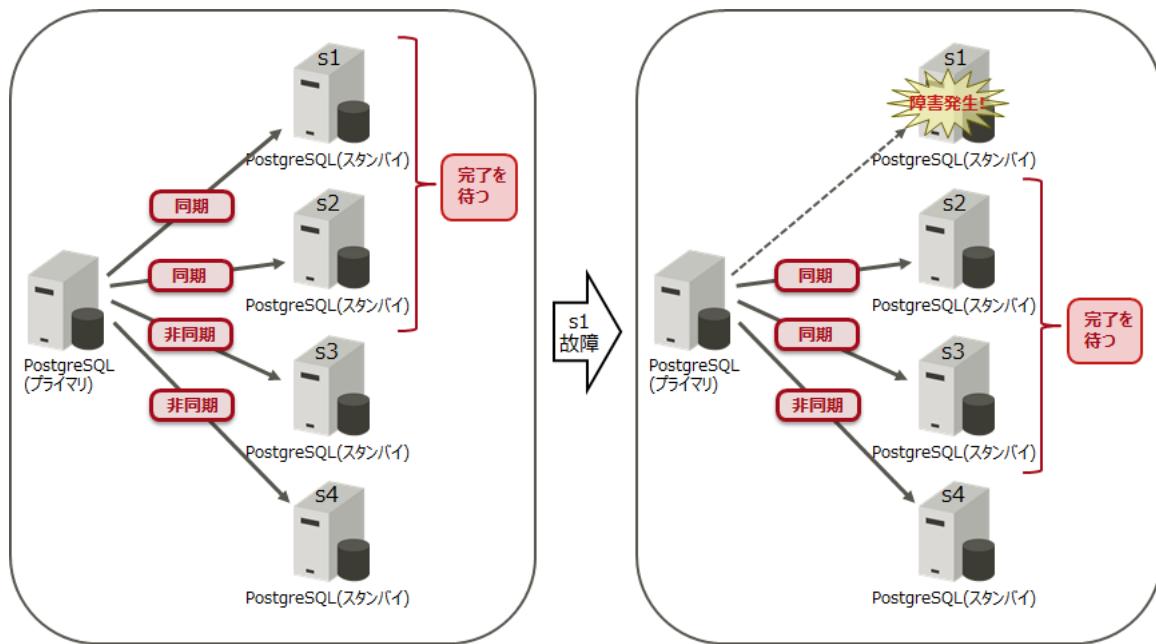
3.2.2 同期/非同期の設定のポイント

同期 / 非同期の設定、および、同期レベルの設定は、`postgresql.conf` のパラメーターで行います。それぞれの設定方法について説明します。

同期 / 非同期の設定 (`synchronous_standby_names`)

同期の設定は、プライマリーサーバーの `postgresql.conf` の `synchronous_standby_names` パラメーターで行います。スタンバイサーバーが複数ある場合は、同期する対象サーバー、`COMMIT` 待ちの優先順などを柔軟に指定できます。このパラメーターで指定されないスタンバイは非同期になります。

設定値の例	概要
<code>'FIRST 2 (s1,s2,s3)'</code>	同期レプリケーションを行うスタンバイサーバーのリストを指定します。「 <code>FIRST n (リスト)</code> 」や「 <code>ANY n (リスト)</code> 」のように、リスト中のサーバーから同期スタンバイを選ぶ方法も指定できます。スタンバイサーバーが、 <code>s1</code> 、 <code>s2</code> 、 <code>s3</code> 、 <code>s4</code> と稼働していた場合、左記例のように指定すると、 <code>s1</code> と <code>s2</code> が同期レプリケーションとなります。この 2 つのスタンバイサーバーの完了を待ってから、プライマリーサーバーは <code>COMMIT</code> します。 <code>s3</code> は、 <code>s1</code> または <code>s2</code> が故障した場合、同期レプリケーションとなります。 <code>s4</code> は、非同期レプリケーションとなります。



図：「`synchronous_standby_names = 'FIRST 2 (s1,s2,s3)'`」設定している場合

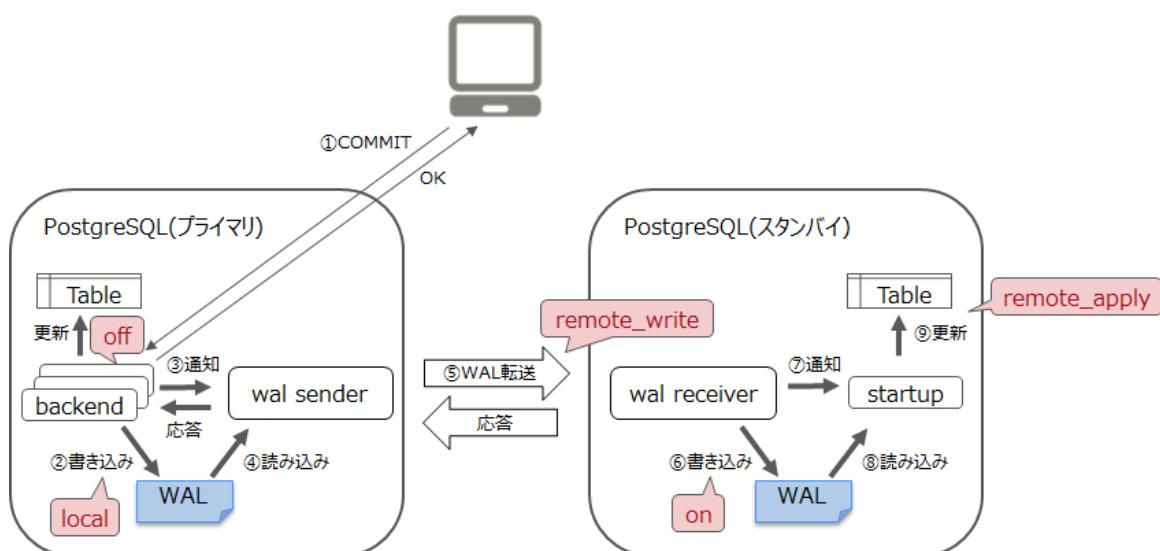
ポイント

`synchronous_standby_names` パラメーターに設定する名前と、スタンバイサーバーの `recovery.conf` の `primary_conninfo` パラメーターの `application_name` に設定する名前は、一致させてください。

同期レベルの設定 (`synchronous_commit`)

スタンバイサーバーの同期レベルの設定は、プライマリーサーバーの `postgresql.conf` の `synchronous_commit` パラメーターで行います。設定できる値と、それぞれの概要について説明します。表中の「保証する範囲」における丸付き数字の番号は表直下の図中の番号を意味しています。

同期レベル	設定値	概要	保証する範囲
完全同期	remote_apply	スタンバイサーバーでの WAL の適用 (データ更新) が終わり、スタンバイサーバー上の参照が可能になってから COMMIT の応答を返します。データの同期が完全に保証されているため、データの最新性を求める参照業務を負荷分散する場合に適しています。	① から ⑨
同期	on (デフォルト)	スタンバイサーバーでの WAL の書き込みが終わってから COMMIT の応答を返します。性能と信頼性のバランスが一番取れています。	①から⑥
準同期	remote_write	スタンバイサーバーへの WAL の転送が終わってから COMMIT の応答を返します。	①から⑤
非同期	local	プライマリーサーバーの WAL の書き込みが終わってから COMMIT の応答を返します。	①から②
非同期	off	プライマリーサーバーの WAL の書き込みが終わるのを待たずに COMMIT の応答を返します。本パラメーターの設定はお勧めできません。	①



PostgreSQL のストリーミングレプリケーションの仕組みと構成のポイントについて解説しました。ストリーミングレプリケーションの運用をするための機能のポイントについては、「ストリーミングレプリケーション ~ 運用のための機能 ~」を参照してください。

【付録】

ストリーミングレプリケーション機能を利用する場合に必要なパラメーターについて説明します。ここでは、同期レプリケーションおよび WAL をアーカイブする場合を例にしています。

プライマリーサーバー側の設定

ファイル名	パラメーター	概要
postgresql.conf	listen_addresses	利用可能なすべての IP インターフェイスに対応する「*」を設定します。
	wal_level	「replica (デフォルト)」を設定します。
	max_wal_senders	「スタンバイサーバーの数+1」を指定します。ただし、max_connections を超える値は設定できません。
	max_connections	データベースサーバーに同時接続する最大数を設定します。 max_wal_senders より大きな値を設定します。
	wal_keep_segments	pg_wal ディレクトリーに保持しておくファイルセグメント数の最小値を設定します。アーカイブしない場合は、少し増やした値を設定します。 参考：PostgreSQL 13 からパラメータ名が wal_keep_size に変更され、ファイルセグメント数指定からサイズ指定に変わりました。
	wal_sender_timeout	wal receiver プロセスが異常な状態になったと判断する時間を設定します。
	synchronous_standby_names	同期レプリケーションをするスタンバイサーバーを設定します。非同期レプリケーションの場合は設定不要です。 例：'s1'
	synchronous_commit	スタンバイサーバーの同期レベルを設定します。データの最新性を求める場合は「remote_apply」を、性能と信頼性を求める場合は「on」を設定します。
	archive_mode	「on」を設定します。
	archive_command	WAL をアーカイブする際のコマンドを設定します。 例：'cp %p /mnt/server/archivedir/%f'
pg_hba.conf	DATABASE 列に「replication」を設定します。 例：host replication postgres スタンバイサーバーのアドレス password	

スタンバイサーバー側の設定

ファイル名	パラメーター	概要
postgresql.conf	hot_standby	「on (デフォルト)」を設定します。
	wal_receiver_timeout	wal sender プロセスが異常な状態になったと判断する時間を設定します。
recovery.conf	standby_mode	「on」を設定します。
	primary_conninfo	プライマリーサーバーへの接続情報を設定します。 application_name と、プライマリーサーバーの postgresql.conf の synchronous_standby_names には、同じ名前を設定します。 例：'host=プライマリーサーバーのアドレス port=5432 user=postgres password=pass application_name=s1 '

ポイント

障害時に、スタンバイサーバーをプライマリーサーバーにする場合に備え、プライマリーサーバーに設定したパラメーターは、スタンバイサーバーでも事前に設定しておくことを推奨します。

2025年10月14日