PostgreSQL の災害対策

災害対策センターの自動フェイルオーバーで広域災害に備える

－富士通の技術者に聞く！PostgreSQL の技術－

近年、台風や豪雨、地震など想像をはるかに超える規模の自然災害が増えています。企業は、万が一、このような災害に直面したときに「事業が継続できるのか」、また、「損害を最小限に食い止めて早期復旧できるか」などを考慮した対策を講じておく必要があります。FUJITSU Software Enterprise Postgres（以降、Enterprise Postgres）では、バージョン 10 で提供された災害対策運用のための機能が今回のバージョン 12 でさらに強化されました。
本特集では、機能の開発担当である「二宮 大介」が、機能をわかりやすく解説するとともに、機能開発への思いや将来に向けた展望について語ります。

二宮 大介 Daisuke Ninomiya
富士通株式会社　ソフトウェアプロダクト事業本部 データマネジメント事業部
専門分野：データベース
入社以来、データベースの開発・保守に携わり、ユーティリティーコマンド、データベース定義、GUI を担当。現在は、PostgreSQL をベースとした「FUJITSU Software Enterprise Postgres」の開発を担当している。

Enterprise Postgres の災害対策を強化

Enterprise Postgres のこれまでの災害対策はどのようなものだったのでしょうか？

二宮

Enterprise Postgres では、PostgreSQL のストリーミングレプリケーションをベースとした災害対策運用に、「ログの順序性保証」という機能を付加して災害対策運用を行えます。「ログの順序性保証」とは、運用センター内で冗長化しているサーバー間の更新ログの送信が完了してから、運用センターと災害対策センター間の更新ログ送信を行うものです。災害対策センターへのログ送信が先行してしまうと、万が一、運用センターのプラマリーサーバーに障害が発生した場合に、運用センターのスタンバイサーバーと災害対策センターとで整合性がとれなくなり、レプリケーションが継続できないという事態が発生しうるのです。「ログの順序性保証」により、確実なレプリケーションが実現できます。

関連コンテンツ

- 「ログの順序性保証」について
  PostgreSQL の災害対策 ～ 災害への備えは万全に! ログの順序性保証で確実なレプリケーション ～

従来の機能でも信頼性の高い災害対策が運用できていたと思いますがいかがでしょうか？

二宮

「ログの順序性保証」は、運用センターで障害が発生した場合でもその影響を受けずに災害対策センターを継続させるための機能です。しかし、災害対策センター内の障害に関して可用性という点で十分ではありませんでした。従来の運用でも、災害対策センター内を冗長化することで、一方のサーバーで障害が発生しても、もう一方のサーバーで運用を継続することは可能でしたが、災害対策センター内での障害発生時のサーバー切り替えにおいて自動フェイルオーバーができないという課題がありました。

自動フェイルオーバーできないとどのような問題がおきるのでしょう？

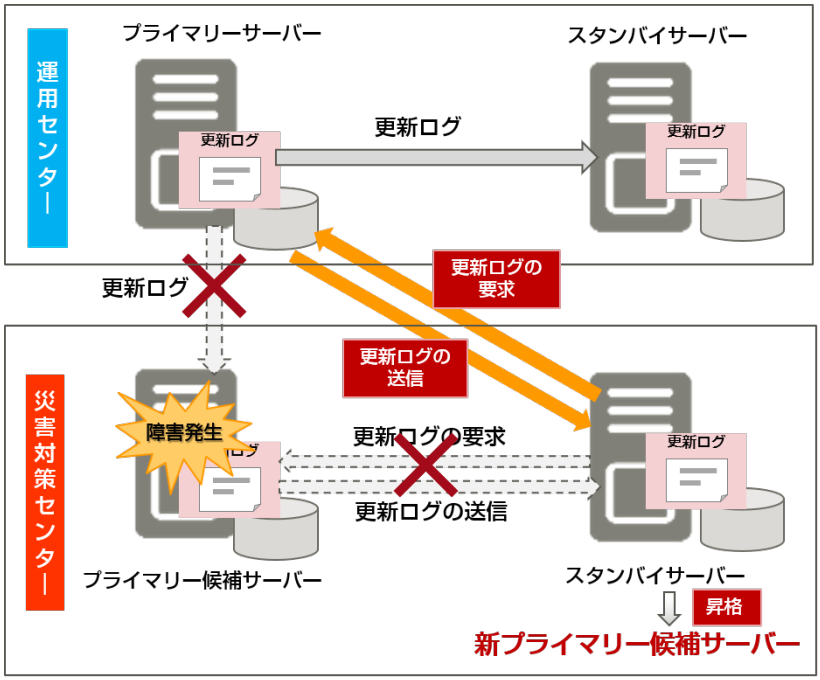
二宮

災害対策センター内の運用サーバーの切り替えを手動で行うことになります。具体的には、災害対策センター内のサーバーの状態監視や運用センターへの接続先の切り替えを利用者が行わなければなりません。障害が発生してもすぐに検出できなかったり、手動でのサーバー切り替えに時間がかかるといった問題があります。また、手動での切り替え中に運用センターが被災した場合、災害対策センターには運用センター被災前の最新の更新ログが送信されないため、運用センター被災直前の状態から業務が再開できないといった事態も起こりえます。これでは、十分な可用性を提供できているとは言えないのです。一方、クラスタソフトウェアを導入することで切り替えを自動化することも可能ですが、コストが上がってしまうという課題もあります。そのため、当社のお客様からも災害対策センター内の切り替えを自動化してほしいという要望があがっていました。

そこで Enterprise Postgres 12 でのエンハンスにつながるわけですね

二宮

はい、Enterprise Postgres 12 では、旧バージョンで利用者が行っていた災害対策センターのサーバーの状態監視を Enterprise Postgres が行うようにしました。また、災害対策センターのプライマリー候補サーバーで障害が発生した場合に、自動フェイルオーバーする機能を提供しました。自動フェイルオーバーでは、Enterprise Postgres が「災害対策センターのスタンバイサーバーを新プライマリー候補サーバーに昇格させる」、また、「更新ログの要求先を旧プライマリー候補サーバーから運用センターのプライマリーサーバーに切り替える」を行います。これらのエンハンスにより、従来と比べて、災害対策センター内のサーバー切り替えが短時間で行えるようになりました。



自動フェイルオーバーのメリットは他にもありますか？

二宮

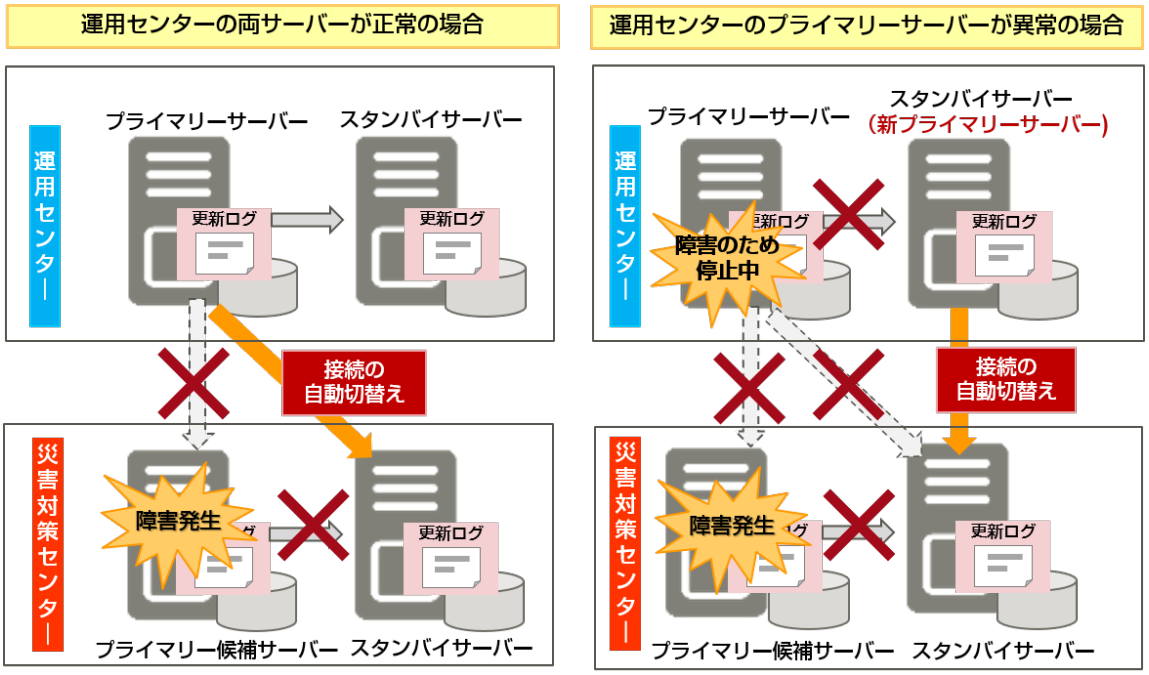
運用センターからの更新ログの受信先を自動で切り替えることができるので、災害対策センターのスタンバイサーバーに対する更新ログの反映漏れなどが発生しません。運用センターとのレプリケーションが確実に継続できます。

災害対策センター内での自動フェイルオーバーの仕組み

それでは災害対策センター内での自動フェイルオーバーをどのように実現しているのか教えてください。

二宮

災害対策センター内の自動フェイルオーバーでは、災害対策センターのスタンバイサーバーが新プライマリー候補サーバーに昇格し、レプリケーションの接続元が自動的に切り替わります。通常運用時は、スタンバイサーバーの接続元はプライマリー候補サーバーですが、運用状況によって運用センターのプライマリーサーバーまたはスタンバイサーバー（新プライマリーサーバー）のどちらかに接続元を切り替えます。



2 つのパターンが想定されるのですね。

二宮

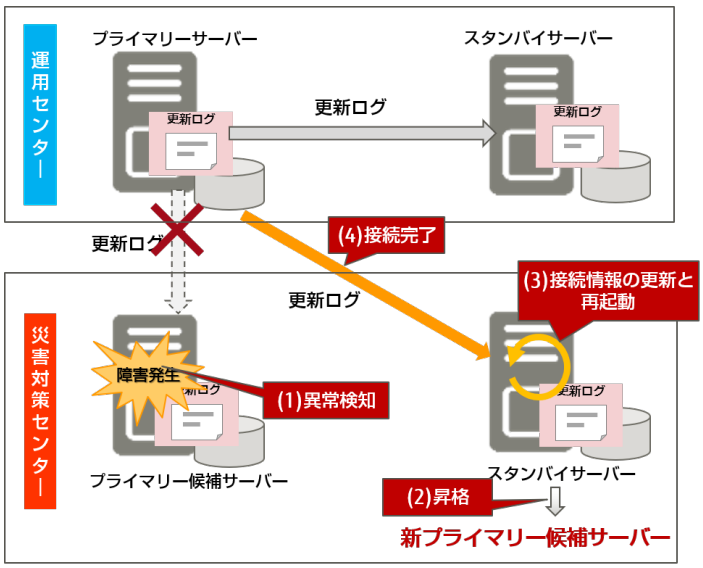
はい、しかし、どちらの場合でも、Enterprise Postgres が接続元を自動で判定するので、利用者が接続を意識する必要はありません。

具体的にはどのような仕組みなのでしょうか？

二宮

「災害対策センターのスタンバイサーバーの接続情報の書き換え」と「運用センターのプライマリーサーバーへの自動接続」の 2 つのフェーズを実装しています。まず、「災害対策センターのスタンバイサーバーの接続情報の書き換え」ですが、Enterprise Postgres の動作環境ファイルの 1 つである「サーバー定義ファイル」に自動フェイルオーバーの接続に関する 2 つのパラメーターを運用前にあらかじめ指定しておきます。1 つは、災害対策センターのスタンバイサーバーの接続元を運用センターのプライマリーサーバーに書き換えるためのパラメーターで、もう 1 つは災害対策センターのプライマリー候補サーバーの接続元をスタンバイサーバーに書き換えるためのパラメーターです。Enterprise Postgres は、災害対策センターのプライマリー候補サーバーの異常を検知した際に、スタンバイサーバーを新プライマリー候補サーバーに昇格させます。そして、このパラメーターの値を利用して、postgresql.conf の primary_conninfo パラメーター（スタンバイサーバーがプライマリーサーバーに接続するためのパラメーター）に指定する、災害対策センター内の各サーバーの接続情報を書き換えます。

なるほど、そして、次のフェーズである「運用センターのプライマリーサーバーへの自動接続」につながるわけですね？



二宮

はい、そうです。Enterprise Postgres は、パラメーターの書き換えが完了したら、災害対策センターの新プライマリー候補サーバーのインスタンスを再起動します。postgresql.conf の primary_conninfo パラメーターの接続文字列である target_session_attrs に、プライマリーサーバーへの接続を示す“read-write”を指定しておくことで、再起動のタイミングで、災害対策センターの新プライマリー候補サーバーは運用センターのプライマリーサーバーに自動的に接続されるという仕組みです。もちろん、運用センターのプライマリーサーバーとスタンバイサーバーが入れ替わっている場合でも、正しい接続先を認識して自動的に接続します。

この仕組みにより、運用センターと災害対策センター間の接続性が向上し、業務継続性が保証されるということですね。この機能を開発、実装するにあたり、こだわったことなどありますか？

二宮

災害対策センターの自動フェイルオーバーの運用についてですが、ほぼ運用センターと同じ操作で行えるようにしました。災害対策センター用のコマンドなどは提供していません。運用センターと災害対策センターは、被災の状況によっては入れ替わることもあり、センター間での運用操作の差異をなくすことで、利用者の誤操作防止や利便性の向上につながると考えたためです。

使いやすさの面も考慮されているんですね。最後に今後の取り組みなどありましたら教えてください。

二宮

現状、センター間の昇格は自動化されていません。たとえば、運用センターが被災した場合には、災害対策センターを運用センターに昇格させて業務を引き継ぐ必要がありますが、現状、その切り替え操作は利用者による手動での操作となっています。今後は、センター間での昇格を自動化することで更なる信頼性と高可用性の向上に取り組んでいきたいと考えています。

さまざまな場面での業務継続性を想定されているんですね。ミッションクリティカルを追求する Enterprise Postgres のこだわりを感じます。Enterprise Postgres の今後の進化が楽しみです。ありがとうございました。

2020 年 10 月 16 日