髙橋 敏行
アマゾン ウェブ サービス ジャパン
パートナーソリューションアーキテクト
2020/2/17作成

AWS INNOVATE 2020 オンラインカンファレンス

「Amazon.com における Oracle データベースの移行」

のクイズおよび解答

AWS INNOVATE 2020のセッションの視聴およびアンケートにご記入頂きありがとうございます。本資料が「Amazon.com における Oracle データベースの移行」で出題されたクイズの解答になります。

問題：データベースの移行において、アプリケーション毎の段階的な移行は重要な要素になります。今回ご紹介した Amazon.com でも、ダウンタイムを減らして、段階移行が出来るようにアーキテクチャーの工夫を行っています。具体的には、アプリケーション毎にデータベースエンジンを選んで接続できるようにしています。どのようなアーキテクチャーを採用しているでしょうか？

解答：Amazon.comでは、セッション内の「Oracle から Amazon RDS/Aurora PostgreSQL へ移行」でお伝えしたアーキテクチャー以外にも別のチャレンジとして、ダウンタイムなしでサービスの移行を行うために作成したリファレンスアーキテクチャがあります。 それがデータベース接続を動的に確立するという方法です。どのように実現したのか、以下にご紹介します。

アプリケーションを（システム単位ではなく）モジュール単位で段階的に移行をする事で、ダウンタイムの極小化を実現することができます。また、アプリケーションの移行を行っていく上では、データベースエンジンの変更対応が完了しているアプリケーションモジュールや、対応中のモジュール、これから対応するモジュール等、様々なフェーズがあります。その中で、ダウンタイムの極小化を実現するため、複数のエンジンに接続できる機構を実装しました。

髙橋 敏行
アマゾン ウェブ サービス ジャパン
パートナーソリューションアーキテクト
2020/2/17作成

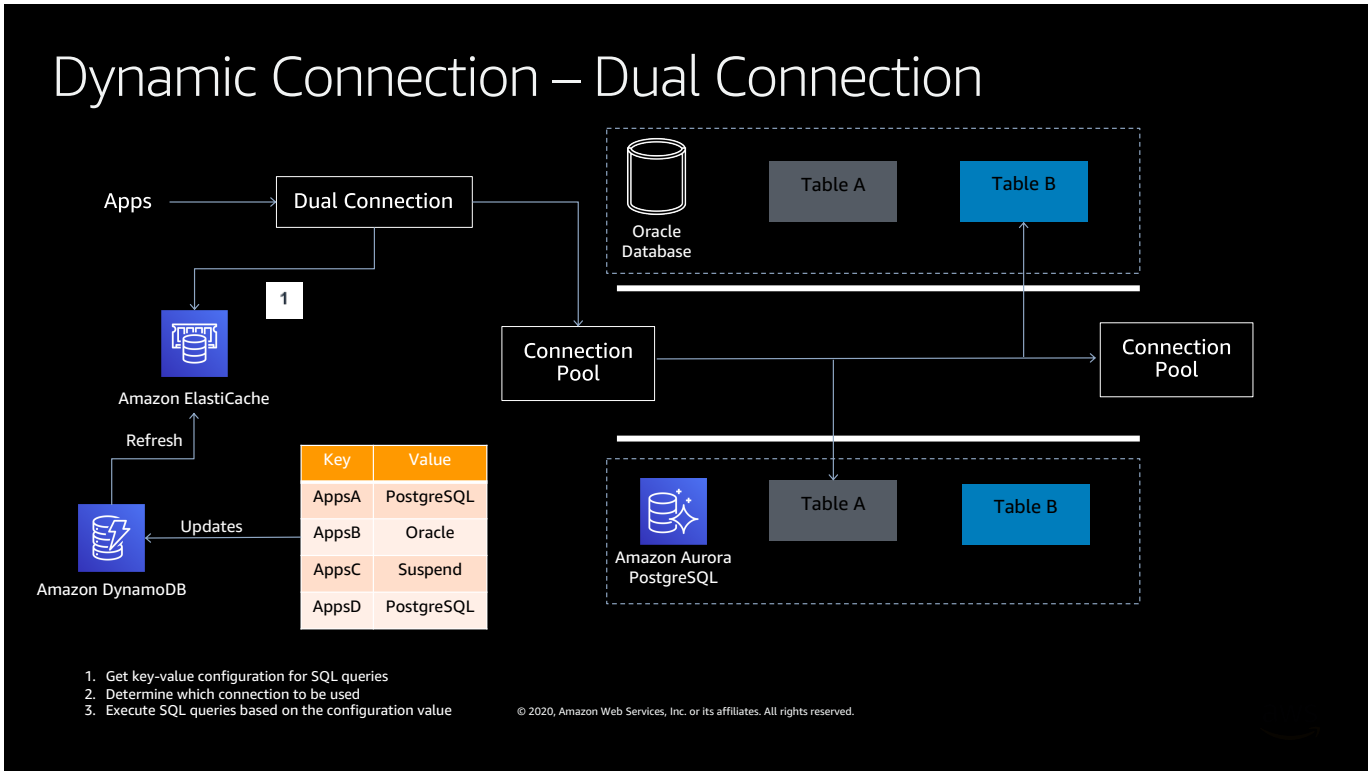

図 1 : Dynamic Connection

Amazon DynamoDBテーブルとキャッシュレイヤーのAmazon ElastiCacheを活用し、各アプリケーションモジュールのマッピング情報をデータベースに保存することで、このアーキテクチャーを実現しています。 （図1）

たとえば、アプリケーションA (AppsA) はPostgreSQLデータベースエンジンにマッピングされ、アプリケーションB (AppsB) はOracleデータベースエンジンにマップされます。 （図1内、Mapping Table）

髙橋 敏行
アマゾン ウェブ サービス ジャパン
パートナーソリューションアーキテクト
2020/2/17作成

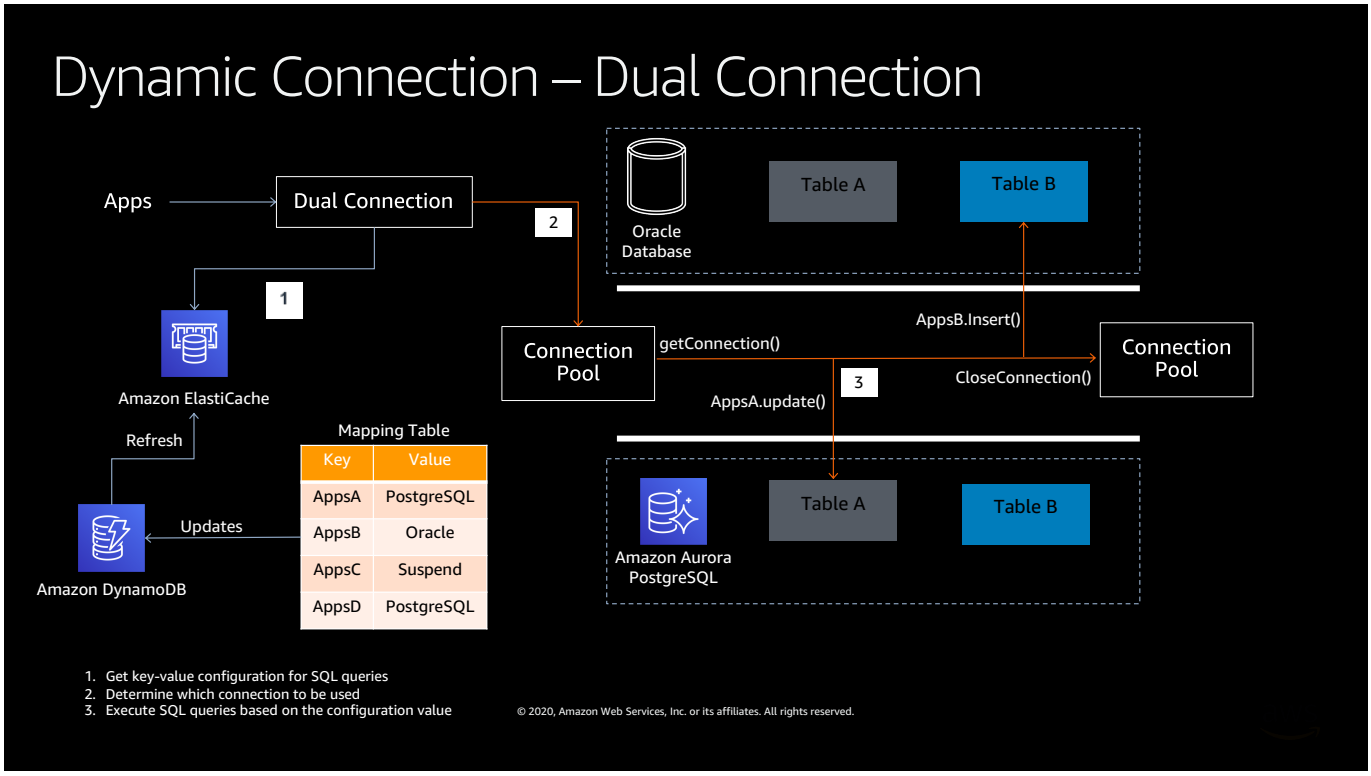

図 2 : Dual Connection

アプリケーションの実行時において、OracleデータベースとPostgreSQLデータベースの両方に接続できるコネクションプール（Dual Connection）を作成しています。（図2） 各アプリケーションモジュールは、Mapping Tableに基づいてSQLを実行する接続を決定します。（図2内、[2] - [3]）

髙橋 敏行
アマゾン ウェブ サービス ジャパン
パートナーソリューションアーキテクト
2020/2/17作成

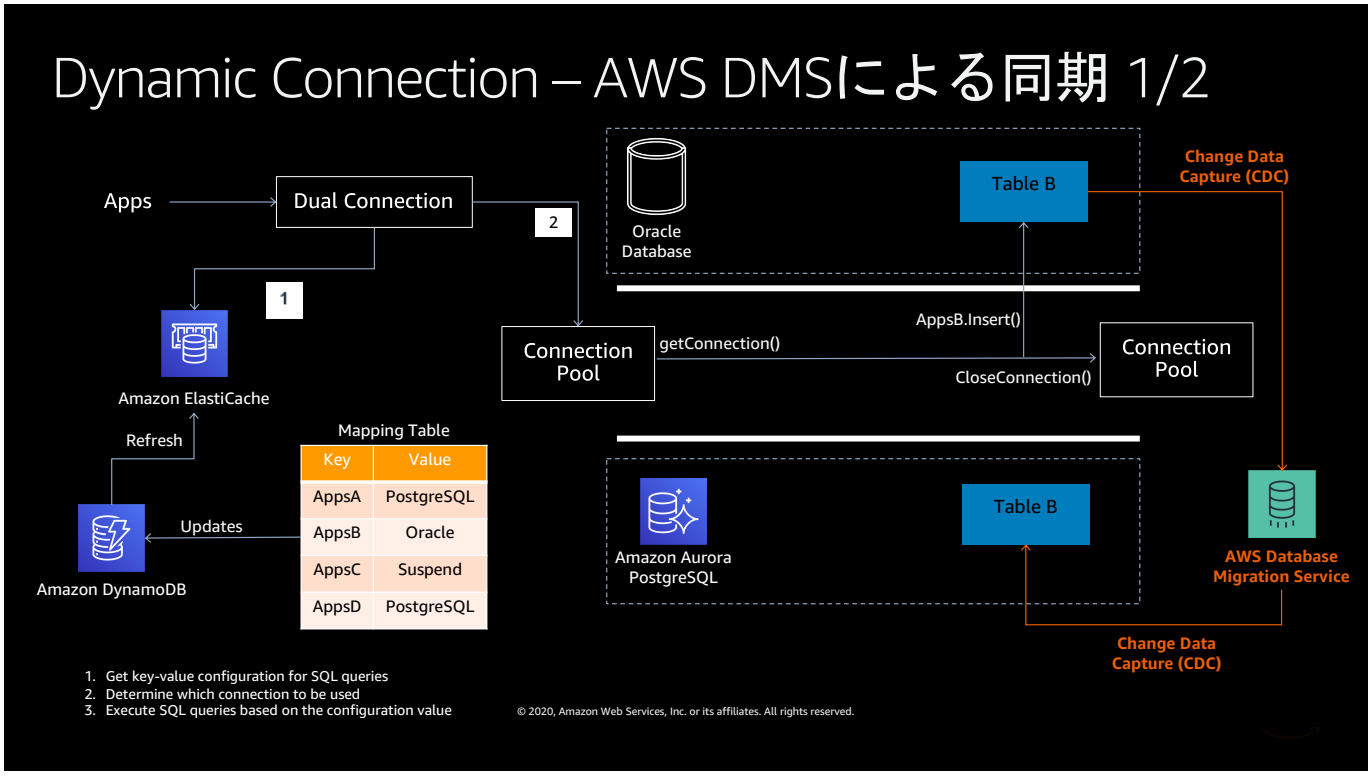

図 3 : AWS DMSによる同期1/2

また、オンプレミスのOracleデータベースと、AWSクラウドのPostgreSQLデータベースのテーブル間でデータを継続的に同期するため、AWS DMSレプリケーション[1]を設定しています。（図3） たとえば 、アプリケーションBがOracleデータベースのテーブルBにレコードを挿入（AppsB.Insert）すると、AWS DMSは変更をキャプチャし（Change Data Capture : CDC）、PostgreSQLデータベースのテーブルBにレプリケートします。これにより、テーブルBはOracleデータベースとPostgreSQLデータベースの両方にあり、同期されています。

[1] AWS Database Migration Service (DMS) : https://aws.amazon.com/jp/dms/

髙橋 敏行
アマゾン ウェブ サービス ジャパン
パートナーソリューションアーキテクト
2020/2/17作成

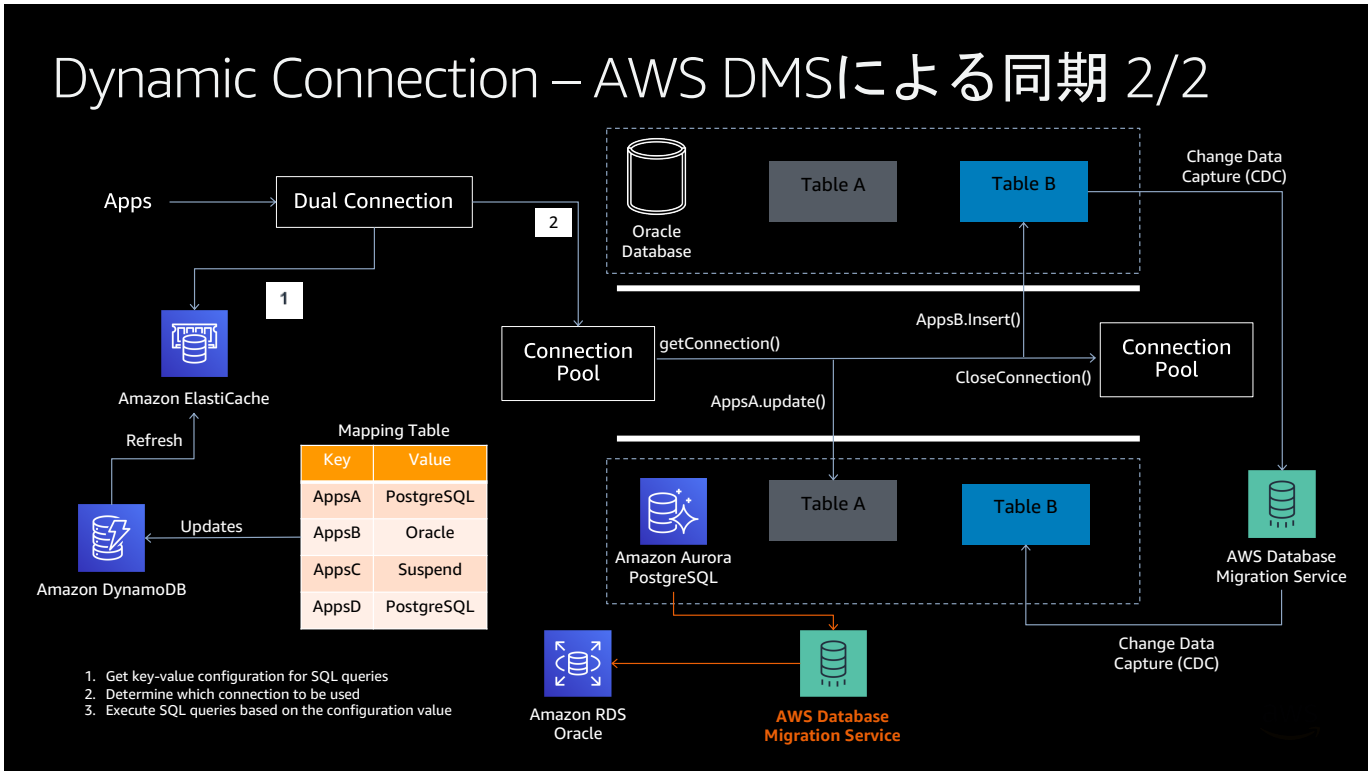

図 4 : AWS DMSによる同期2/2

同様に、 アプリケーションAがテーブルAを更新する場合、AWSクラウドのPostgreSQLデータベースのレコードが更新されます（AppA.update）。また、AWS上にOracle（RDS for Oracle）を構成し、Aurora PostgreSQL からRDS for OracleへDMSレプリケーションを設定しています。この仕組みにより、アプリケーションのカットオーバー準備ができる前のPostgreSQLデータベースに障害が発生した場合にも、RDS for Oracleに接続する事でアプリケーションの継続利用を実現します。

最後までお読みいただきありがとうございました。以上が、移行によるアプリケーションのダウンタイムの極小化を実現するリファレンスアーキテクチャーの1つとなります。今後のご参考になりましたら幸いです。

ご参考リンク

- Amazon Innovator [Amazon.com on AWS] : https://aws.amazon.com/jp/solutions/case-studies/innovators/amazon/