eBook

データエンジニアリングのビッグブック

コードサンプルと Notebook を含む
技術ブログ集

databricks

目次

SECTION

01

Databricks を活用したデータエンジニアリング（入門編）

組織は、収益の拡大、顧客体験の向上、効率的な運用、製品やサービスの改善など、ビジネスに関連するさまざまな取り組みにおいて、データが戦略的な資産として果たす価値を認識しています。しかし、こうした取り組みのためのデータへのアクセスや管理は、ますます複雑になってきています。このような複雑さは、データ量とデータタイプの爆発的な増加に伴って生じており、組織が保持するデータの 80% 以上が非構造化および半構造化形式であると推定されています。データの収集が増え続けるなか、データの 73% が分析や意思決定に使用されないままになっています。この割合を減らし、より多くのデータを利用できるようにするため、データエンジニアリングチームは、効率的かつ確実にデータを提供するデータパイプラインの構築を担っています。しかし、このような複雑なデータパイプラインを構築するプロセスには、次に挙げるような困難が伴うことも実情です。

- データレイクにデータを取り込むために、反復的なデータ取り込み作業を手作業で行う必要があり、膨大な時間を費やすことになります。
- データプラットフォームは常に変化しているため、複雑なスケーラブルインフラの構築と維持、再構築に時間がかかります。
- リアルタイムデータの重要性が高まるなか、低遅延のデータパイプラインが必要であり、その構築と維持がさらに困難です。
- 全てのパイプラインが書き込まれた後、データエンジニアは常に性能に注目し、パイプラインとアーキテクチャを調整して SLA を満たす必要があります。

Databricks をどのように活用するか

Databricks レイクハウスプラットフォームは、データエンジニアがデータの取り込み、変換、処理、スケジューリング、配信を行うためのエンドツーエンドのデータエンジニアリングソリューションを利用できるようにするものです。レイクハウスプラットフォームは、パイプラインの構築と維持、ETL ワークロードの実行といった複雑な作業をデータレイク上で直接自動化するため、データエンジニアは品質と信頼性に集中し、価値あるインサイトを導き出すことができます。

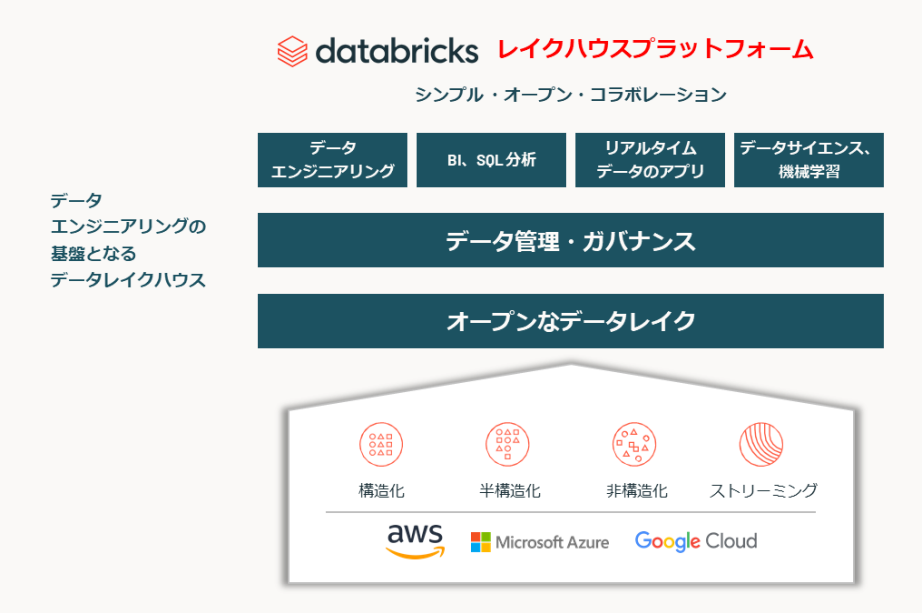


図 1： Databricks レイクハウスプラットフォームは、データ、分析、AI を単一のプラットフォームに統合し、あらゆるデータユースケースに対応します。

Databricks のデータエンジニアリングにおける競争優位性

データエンジニアは、レイクハウスアーキテクチャを簡素化することで、データパイプラインの構築にエンタープライズグレードでエンタープライズ対応のアプローチを必要としています。データエンジニアリングソリューションチームが成功するためには、以下の8つの主要な差別化機能を採用する必要があります。

連続またはスケジュールされたデータ取り込み

自動進化するスキーマでペタバイト級のデータを取り込むことができるため、データエンジニアは、分析、データサイエンス、または機械学習のために、高速で信頼性が高く、スケーラブルで自動化されたデータを提供することができます。これには以下が含まれます。

- Kafka、DBMS、NoSQL などのファイルやストリーミングソースから到着したデータのインクリメンタルかつ効率的な処理
- 構造化および非構造化データ形式に対するスキーマの自動推論とカラムの変更の検出
- 到着したデータを人手を介さず自動的かつ効率的に追跡
- データ列の救済によるデータ損失の防止

宣言型 ETL パイプライン

データエンジニアは、開発の時間と労力を削減し、代わりに SQL や Python を使用してデータパイプライン内のビジネスロジックとデータ品質チェックの実装に集中することができます。これは次のような方法で実現できます。

- 意図駆動型の宣言型開発により「どのように」解決するかを簡略化して「何を」解決するかを定義
- 高品質なリネージの自動的な作成によるデータパイプライン全体でテーブルの依存関係の管理
- 依存関係の欠落やシンタックスエラーの自動チェックとデータパイプラインの復旧の管理

データ品質の検証およびモニタリング

データレイクハウス全体のデータの信頼性を向上させ、データチームが下流工程で使用する情報を自信を持って信頼できるようにするために、次のことを行います

- パイプライン内のデータ品質と整合性制御、データの期待値を定義
- 事前定義されたポリシー（フェイル、ドロップ、アラート、検疫）によるデータ品質エラーに対処
- データパイプライン全体について取得、追跡、報告されるデータ品質指標を活用

フォールトトレラントと自動回復

過渡的なエラーを処理し、パイプラインの運用中に発生する最も一般的なエラー状態から、高速でスケーラブルな自動リカバリにより回復します。

- データの状態を一貫して回復するフォールトトレラントなメカニズム
- チェックポイントにより、ソースからの進捗を自動的に追跡する機能
- データパイプラインの状態を自動的に回復、復元する機能

データパイプラインの可観測性

データフローグラフダッシュボードからデータパイプライン全体の状態を監視し、性能、品質、レイテンシのエンドツーエンドのパイプラインの健全性を視覚的に追跡できます。次のデータパイプラインの観測可能な機能があります。

- 高品質で忠実度の高い系統図でインパクト分析のデータフローを可視化
- データパイプラインの性能とステータスを行単位で詳細に記録
- データパイプラインジョブの継続的なモニタリングによる運用継続の確認

バッチおよびストリームデータ処理

データエンジニアは、複雑なストリーム処理の知識やリカバリーロジックの実装を必要とせず、コストコントロールしながらデータレイテンシを調整できます。

- データパイプラインのワークロードを、自動プロビジョニングされた弾力性のある Apache Spark™ ベースの計算クラスタで実行し、スケールと性能を実現
- ジョブの並列化とデータ移動の最小化を可能にする性能最適化クラスタ

自動デプロイメントと自動オペレーション

データパイプラインの展開とロールバックを簡単かつ自動的に行い、ダウンタイムを最小限に抑えることで、分析および機械学習のユースケースにおいて、信頼性と予測性の高いデータ配信を実現します。以下のような利点があります。

- データの継続的な配信のための完全でパラメータ化された自動デプロイメント
- 主要クラウドプロバイダにおけるデータパイプライン展開のエンドツーエンドのオーケストレーション、テスト、モニタリング

スケジュールされたパイプラインとワークフロー

データおよび機械学習パイプラインのデータ処理タスクのオーケストレーションを、シンプル、明確、かつ信頼性の高い方法で実現します。複数の非インタラクティブなタスクを DAG（Directed Acyclic Graph）として Databricks コンピュートクラスタで実行することが可能です。

- Databricks の UI と API を使用して、DAG 内のタスクのオーケストレーション
- UIやAPIを介して複数のタスクをジョブで作成・管理し、監視のためのメールアラートなどの機能を搭載
- Databricks 外部の API を持つあらゆるタスクをあらゆるクラウドでオーケストレート

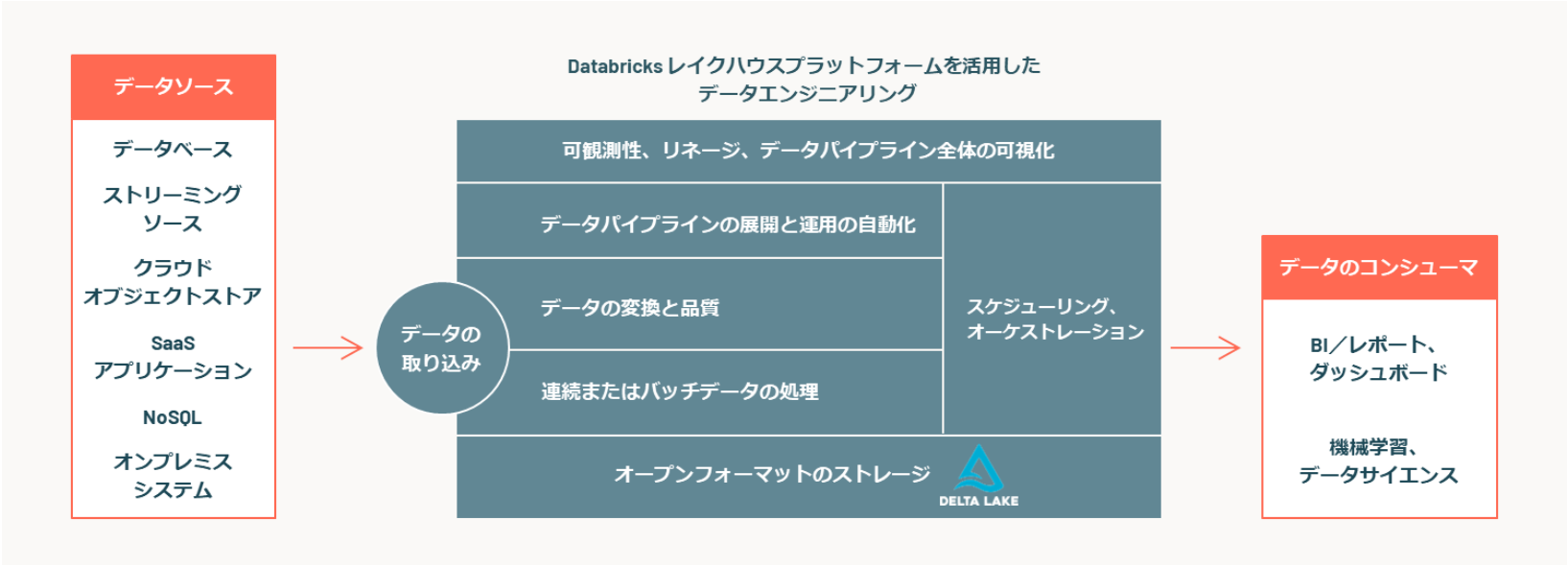


図 2：Databricks のリファレンスアーキテクチャに基づくデータエンジニアリング

結論

組織がデータドリブンを目指すなか、データエンジニアリングは成功のためのカギとなっています。高信頼性データの供給のため、データエンジニアが ETL ライフサイクルを手動で開発・管理するのは効率的ではなく、ETL 開発の簡素化、データの信頼性および運用効率の向上を可能にするスケーラブルな手段が求められています。

前述の 8つの差別化機能は、あらゆるデータの依存関係を自動化し、維持し、品質管理機能による監視と自動回復によりパイプライン運用の詳細を可視化することで、ETL ライフサイクルの管理を簡素化しています。データエンジニアリングチームは、分析、データサイエンス、機械学習のためのデータを供給するバッチおよびストリーミングに対し、SQL または Python のみを使用して信頼性の高いエンドツーエンドの本番対応データパイプラインを容易かつ迅速に構築できるようになりました

ユースケース

次のセクションでは、実例をもとにしたデータエンジニアリングのエンドツーエンドのユースケースに関するベストプラクティスを説明します。データの取り込みからデータ処理、分析、機械学習まで、未加工データを実用的なデータに変換する方法を学びます。データセットとコードサンプルを利用して、実際に Databricks レイクハウスプラットフォームでデータライフサイクルのあらゆる側面を探求できます。

SECTION

02 Databricks レイクハウスプラットフォームにおける実例ユースケース

- データレイクハウスによるリアルタイム POS 分析
- CrowdStrike Falcon イベントのためのサイバーセキュリティレイクハウスの構築
- 最新のデータレイクハウスでヘルスケアデータの可能性を最大化
- 規制レポートの提出における即時性と信頼性の確保
- Databricks レイクハウスプラットフォームを利用した AML ソリューション
- ソリューションアクセラレータ：ゲームにおける有害性の検知
- 導入事例：ノースウェスタン・ミューチュアル（Northwestern Mutual）— レイクハウスによる変革の推進
- 主要な クラウドと 50 以上のリージョンにまたがる Databricks レイクハウスの構築

SECTION 2.1

データレイクハウスによるリアルタイムPOS分析

投稿者：Bryan Smith、 Rob Saker

2021年9月9日

商品供給の減少や倉庫のキャパシティーの低下といったサプライチェーンの混乱と、シームレスなオムニチャネル体験に対する消費者の期待の急速な変化が相まって、小売業者はデータを活用した業務管理の方法を見直す必要に迫られています。パンデミック以前は、71% の小売業者が、オムニチャネルの目標を達成するための最大の障害として、在庫のリアルタイムの可視性の欠如を挙げていました。パンデミックにより、オンラインと店舗を統合した体験に対する需要が高まり、小売業者は正確な商品在庫を提示し、注文の変更を即座に管理しなければならないというプレッシャーがさらに高まりました。リアルタイムの情報へのアクセスを向上させることが、新しい時代の消費者の要求に応える鍵になります。

このブログでは、リテール向けのリアルタイムデータの必要性と、POS データのリアルタイムストリーミングを大規模に動かすための課題を、データレイクハウスで克服する方法について説明します。

POS システム

POS システムは、小売業者とお客様の間の商品やサービスのやりとりを記録する店舗インフラの中心的な役割を担ってきました。このやりとりを維持するためにPOSは通常、製品のインベントリを追跡し、単位数が限界レベルを下回ると補充を促進します。

POS は店舗運営において重要です。また、販売と在庫管理の記録システムとして、そのデータへのアクセスはビジネスアナリストにとって重要な関心事です。

従来は、店舗とオフィス間の接続が限られていたため、POS システム（端末インターフェースだけでなく）は物理的に店舗内に存在していました。オフピーク時には、これらのシステムは電話をかけてサマリーデータを送信することがあります。このデータをデータウェアハウスに統合すると、小売オペレーションの性能に関する1日前のビューが提供されますが、次の日のサイクルが始まるまで、データは陳腐化していきます。

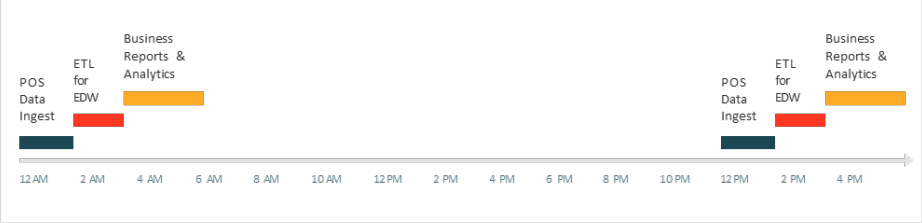


図1：従来のバッチ指向のETLパターンによる在庫の可用性

昨今の接続性の向上により、多くの小売業者がクラウドベースの集中型 POS システムに移行し、また多くの小売業者が店舗内のシステムと企業のバックオフィスとの間でほぼリアルタイムの統合を開発しています。ほぼリアルタイムで情報を入手できるため、小売業者は商品の在庫状況の予測を継続的に更新することができます。企業は、前日までのインベントリ状態の知識をもとにオペレーションを管理するのではなく、現在のインベントリ状態の知識をもとにアクションを起こすことができるようになりました。

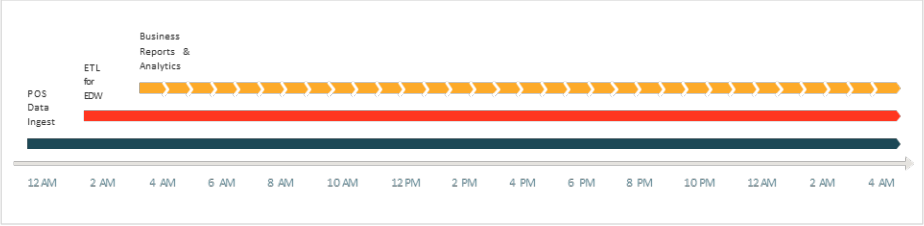


図 2：ストリーミング ETL パターンによるインベントリの有効活用

ほぼリアルタイムの知見

店舗の動向をリアルタイムに近い形で把握することは非常に重要ですが、毎晩の処理から継続的な情報のストリーミングへの移行は、これまでとは異なるデータ処理のワークフローを設計しなければならないデータエンジニアだけでなく、情報を消費する側にも特別な課題をもたらします。この記事では、このジャーニーを最近開始したお客様から学んだ教訓を紹介し、レイクハウスでパターン利用できる主要なパターンとそのパターンから得られる機能の成果を検証します。

教訓 1：スコープを慎重に検討する

POS システムは、多くの場合、販売やインベントリ管理だけに限定されるものではありません。その代わり、決済処理、店舗クレジット管理、請求書発行、発注、ポイントプログラム管理、従業員のスケジュール管理、タイムトラッキング、さらには給与計算まで、幅広い機能を提供することができ、まさに店舗機能の万能ツールといえます。

その結果、POS 内に収容されるデータは、通常、大規模で複雑なデータベース構造に分散しています。運が良ければ、POS ソリューションがデータアクセスレイヤーを利用できるようになり、より解釈しやすい構造でこのデータにアクセスできるようになります。しかし、そうでない場合、データエンジニアは不透明なテーブルを整理して、価値があるデータとそうでないデータを判断しなければなりません。

データの公開方法に関係なく、従来のガイダンスが当てはまります。ソリューションの説得力のあるビジネス上の正当性を特定し、それを使用して最初に消費する情報資産の範囲を制限します。このような正当化は、特定のビジネス課題に対処する任務を負い、よりタイムリーな情報の利用が成功に不可欠であると考える強力なビジネススポンサーからなされることが多いのです。

ここで、多くの小売企業が抱える重要な課題、オムニチャネルソリューションの実現について考えてみましょう。BOPIS（オンライン注文店頭受取）やクロスストアの取引を可能にするこうしたソリューションでは、店舗の在庫情報がある程度正確に把握されていることが前提になります。もし、最初にこの1つのニーズに絞れば、監視、分析システムに必要な情報は劇的に少なくなります。リアルタイムの在庫管理ソリューションが提供され、ビジネスにおいて価値が認められると、プロモーションのモニタリングや不正行為の検出など、他のニーズにも対応できるよう範囲を広げ、繰り返し利用する情報資産の幅を広げていくことができます。

教訓 2：データ生成のパターンと時間感度に合わせた送信の調整

プロセスによって、POS 内で生成されるデータは異なります。販売取引は、関連するテーブルに新しいレコードが追加された痕跡を残す可能性が高く、返品は、過去の販売記録の更新、新しい販売記録の挿入、返品専用の構造体への新しい情報の挿入など、複数の経路をたどる可能性があります。イベント固有の情報が POS 内のどこにどのように配置されているかを正確に把握するには、ベンダーの文書、部族の知識、さらには独自の調査作業が必要な場合があります。

これらのパターンを理解することで、特定の種類の情報に対するデータ送信戦略を構築できます。高頻度、細粒度、挿入指向のパターンは、連続ストリーミングに理想的かもしれません。頻度が少なく、大規模なイベントは、バッチ指向のバルクデータ伝送スタイルが最適でしょう。しかし、これらのデータ伝送モードがスペクトルの両端を表す場合、POS によってキャプチャされたほとんどのイベントがその中間のどこかにあることがわかります。

データレイクハウスのデータアーキテクチャの優れた点は、複数のデータ転送モードを並行して採用できることです。また、連続伝送に適したデータについては、ストリーミングを採用することもできます。一括送信に適したデータについては、バッチ処理を実行できます。そして、その中間にあるデータについては、意思決定に必要なデータの適時性に焦点を当て、それが今後の道筋を決定づけるようにすることができます。これらのモードは全て、ETL の実装に対する一貫したアプローチで取り組むことができます。この課題は、ラムダアーキテクチャと呼ばれる初期の実装の多くを妨げていました。

教訓 3：データを段階的に着地させる

店舗内の POS システムから届くデータは、頻度、フォーマット、タイムリーな入手に対する期待度がそれぞれ異なります。レイクハウスで一般的なブロンズ、シルバー、ゴールドのデザインパターンを活用することで、データの初期クレンジング、再フォーマットおよび永続化を、特定のビジネス成果を目的とした複雑な変換と分離できます。

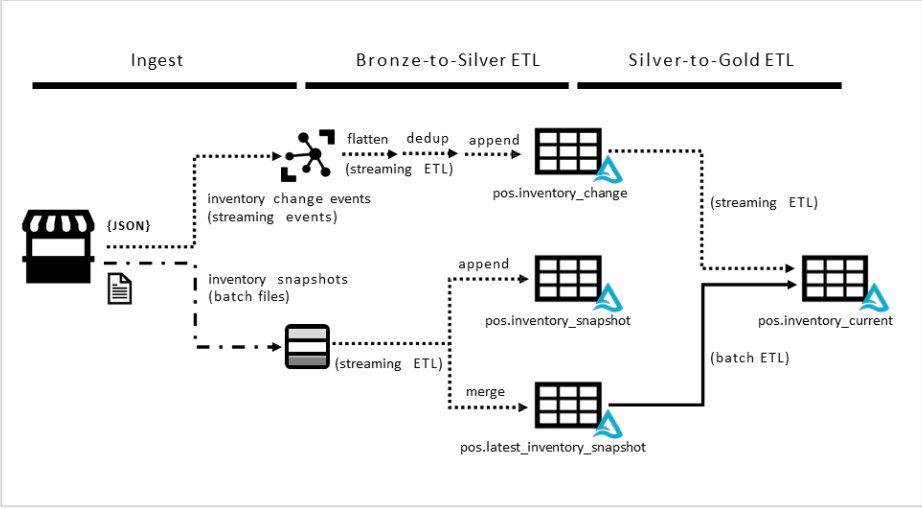


図 3：ブロンズ、シルバー、ゴールドのデータ永続化パターンを活用したインベントリ計算のためのデータレイクハウスアーキテクチャ

教訓 4：エクスペクテーションの管理

ほぼリアルタイムの分析への移行には、組織のシフトが必要です。ガートナーは、ストリーミングデータの分析が日常業務に統合されるようになるまでのストリーミングアナリティクス成熟度モデルを示しています。これは一朝一夕にできることではありません。

その代わりに、データエンジニアは、物理的な店舗からクラウドベースのバックオフィスへのストリーミング配信に固有の課題を認識するための時間が必要です。接続性とシステムの信頼性が向上し、ETL ワークフローがより強固になったことで、データの適時性、信頼性、一貫性がより向上しました。そのためには、システムエンジニアやアプリケーション開発者とのパートナーシップを強化し、バッチ処理のみの ETL ワークフローの時代にはなかったレベルの統合をサポートする必要があります。

ビジネスアナリストは、継続的に更新されるデータ特有のノイズに精通する必要があります。数秒前に実行したクエリが少し違う結果を返した場合など、データセットの診断と検証の方法を学び直す必要があります。日々の集計では見えないデータの問題点を深く認識する必要もあります。そのためには、分析方法と検出されたシグナルへの対応の双方を調整する必要があります。

これらは全て、成熟の最初の数段階に行われるものです。後の段階では、ストリーム内の意味のあるシグナルを検出する能力が、より自動化されたセンスとレスポンスの能力につながるかもしれません。ここでは、データストリームに含まれる最高レベルの価値を解き放ちます。しかし、ビジネスにおいては、これらのテクノロジーに業務を委ねる前に、監視とガバナンスが整備され、実証されなければなりません。

POS ストリーミングの導入

レイクハウスアーキテクチャを POS データに適用する方法を説明するために、ほぼリアルタイムのインベントリを計算するデモワークフローを開発しました。このシステムでは、2 つの別々の POS システムが、販売、在庫補充、シュリンクデータに関連する在庫関連情報を、ストリーミング在庫変更フィードの一部として、オンライン購入、店舗でのピックアップ（BOPIS）取引（一方のシステムで開始され、他方で完了する）を送信すると想定しています。棚にある商品単位の定期的（スナップショット）なカウントを POS で取得し、一括送信します。これらのデータを 1か月間シミュレーションし、10 倍速で再生することで、在庫変動の可視性を高めています。

ETL のプロセス（図 3）は、ストリーミングとバッチの混合技法です。シルバーレイヤーと呼ばれる Delta テーブルに取り込まれた最小限の変換データによる 2 段階のアプローチにより、初期の技術的な ETL アプローチと、現在の在庫計算に必要なビジネス的なアプローチを分離しています。第 2 段階は、従来の構造化ストリーミング機能を使用して実装されています。これは、一般公開に向けて新しい Delta Live Tables 機能で再検討する可能性があります。

このデモでは、データの取り込みに Azure IOT Hubs と Azure Storage を使用していますが、AWS や GCP のクラウドでも適切な技術で代用することで同様に動作させることが可能です。再生可能な ETL ロジックを含む環境のセットアップの詳細は、以下の Notebook に記載されています。

Databricks Notebook の無料お試し

- POS 01：環境設定
- POS 02：データ生成
- POS 03：インジェスト ETL
- POS 04：現在のインベントリ状況

SECTION 2.2

CrowdStrike Falconイベントのための サイバーセキュリティレイクハウスの構築

投稿者：Amero Amare、Arun Pamulapati、Yong Sheng Huang、Jason Pohl

2021年5月20日

エンドポイントデータは、セキュリティチームが脅威の検出や追跡、インシデント調査、コンプライアンス要件を満たすために必要とするものです。データ量は1日あたりテラバイト、1年あたりペタバイトになることもあります。多くの企業は、エンドポイントログの収集、保存、分析に苦労しています。その理由は、このような大量のデータに関連するコストと複雑さにありましたが、その時代は終わりました。

このブログシリーズでは、ペタバイト級のエンドポイントデータを Databricks で運用し、高度な分析によってセキュリティ体制の改善のコスト効率の高い方法を 2 部にわたって紹介します。第 1 部（本ブログ）では、データ収集のアーキテクチャと SIEM（Splunk）との連携について説明します。このブログの最後には、ノートブックが提供され、データを使って分析する準備ができます。第 2 部では、具体的なユースケース、ML モデルの作成方法、自動化されたエンリッチメントと分析について解説します。第 2 部終了時には、エンドポイントデータを利用した脅威の検出と調査のためのノートブックを実装できるようになります。

ここでは CrowdStrike Falcon のログを例に挙げて説明します。Falcon のログにアクセスするには、Falcon Data Replicator（FDR）を使用して未加工のイベントデータを CrowdStrike のプラットフォームから Amazon S3 などのクラウドストレージにプッシュします。このデータは、Databricks レイクハウスプラットフォームを使用して、他のセキュリティ遠隔測定データとともに取り込み、変換、分析、保存できます。

CrowdStrike Falcon のデータを取り込み、Python ベースのリアルタイム検知を適用し、Databricks SQL で履歴データを検索し、Databricks Add-on for Splunk で Splunk などの SIEM ツールからクエリを実行可能です。

CrowdStrikeデータの運用に向けた課題

CrowdStrike Falcon のデータは包括的なイベントログの詳細を提供しますが、複雑で膨大なサイバーセキュリティデータをほぼリアルタイムで取り込み、処理し、費用対効果の高い方法で運用するのは大変な作業です。以下は主な課題の一部です。

- **リアルタイムで大規模なデータ取り込みを実現：**FDR によってほぼリアルタイムでクラウドストレージに書き込まれる、処理済み／未処理の未加工データファイルの把握は困難です。
- **複雑な変換：**データ形式は半構造化されています。各ログファイルの行には、数百種類のペイロードが含まれており、イベントデータの構造は時間の経過とともに変化する可能性があります。
- **データガバナンス：**データは機密性が高いため、必要なユーザーだけにアクセスを制限する必要があります。

- **セキュリティ分析のエンドツーエンドを簡素化：**このような高速・大容量のデータセットに対して、データエンジニアリング、ML、分析を行うためには、スケーラブルなツールが必要です。
- **コラボレーション：**効果的なコラボレーションにより、データエンジニア、サイバーセキュリティアナリスト、ML エンジニアのドメイン専門知識を活用できます。このように、コラボレーションプラットフォームを持つことで、サイバーセキュリティの分析・対応作業の効率を向上させることができます。

その結果、企業内のセキュリティエンジニアは、コストや業務効率の管理に悩まされ、厳しい状況に置かれています。極めて高価なプロプライエタリシステムに縛られることを受け入れるか、スケーラビリティと性能に挑戦しつつ多大な労力を費やして独自のエンドポイントセキュリティツールを構築するかのいずれかかです。

Databricks のサイバーセキュリティ向けレイクハウス

Databricks は、セキュリティチームやデータサイエンティストが効率的かつ効果的に業務を遂行するための新たな希望と、増大するビッグデータや巧妙な脅威といった課題に対処するための一連のツールを提供します。

レイクハウスは、データレイクとデータウェアハウスの利点を備えたオープンアーキテクチャで、データに段階的に構造を追加するマルチホップのデータエンジニアリングパイプラインの構築を簡素化します。マルチホップアーキテクチャの利点は、未加工データを「単一の情報源」とするパイプラインを構築できることです。CrowdStrike の半構造化未加工データは長期保管できます。その後の変換や集約はエンドツーエンドのストリーミング方式で行えます。データの製錬、状況に応じた構造への変換、さまざまなシナリオでのセキュリティリスクの分析、検知が可能です。

- **データの取り込み：**オートローダ（AWS | Azure | GCP） は、CrowdStrike FDR によって新しいファイルが未加工データストレージに書き込まれると同時に、すぐにデータを読み出すのに役立ちます。クラウド通知サービスを活用し、新しいファイルがクラウド上に到着するとインクリメンタルに処理します。また、オートローダは、新しいファイルの通知サービスを自動的に設定してリッスンし、1秒間に数百万ファイルまでスケールアップすることが可能です。
- **ストリーム処理とバッチ処理の統合：**Delta Lake は、データレイクにデータ管理とガバナンスをもたらすオープンなアプローチで、Apache Spark™ の分散計算能力を活用して、膨大な量のデータとメタデータを処理します。Databricks の Delta Engineは、高度に最適化されたエンジンで、1秒間に数百万件のレコードを処理できます。
- **データガバナンス：**Databricks Table Access Control （AWS | Azure | GCP）を使用すると、管理者はユーザーのビジネス機能に基づいてデルタテーブルへの異なるレベルのアクセス権を付与できます。

- **セキュリティ解析ツール：**Databricks SQL は、異常なパターンが検出された場合に自動的にアラートを出す、インタラクティブなダッシュボードの作成を支援します。また、Tableau、Microsoft Power BI、Looker など、採用率の高い BI ツールとの連携も容易に行えます。
- **Databricks Notebook での共同作業：**Databricks のコラボレーション型 Notebook により、セキュリティチームはリアルタイムにコラボレーションを行うことができます。複数のユーザーが同じワークスペース内で多言語のクエリを実行し、ビジュアライゼーションの共有やコメントを行うことで、調査を中断することなく進めることができます。

CrowdStrike Falcon データのための レイクハウスアーキテクチャ

CrowdStrike Falcon データなど、サイバーセキュリティのワークロードには、以下のようなレイクハウスアーキテクチャを推奨しています。オートローダ と Delta Lake は、低コストかつ最小限の DevOps 作業で、クラウドストレージからローデータを読み取り、デルタテーブルに書き込むプロセスを簡素化します。

このアーキテクチャでは、半構造化された CrowdStrike のデータがランディングゾーンのお客様のクラウドストレージにロードされます。その後、オートローダはクラウド通知サービスを利用して、新しいファイルの処理とお客様のブロンズテーブルへの取り込みを自動的に開始し、下流の全てのジョブの単一の真実のソースとして機能するようにします。オートローダは、データの重複処理を防ぐため、チェックポイントを使用して処理済みファイルと未処理ファイルを追跡します。

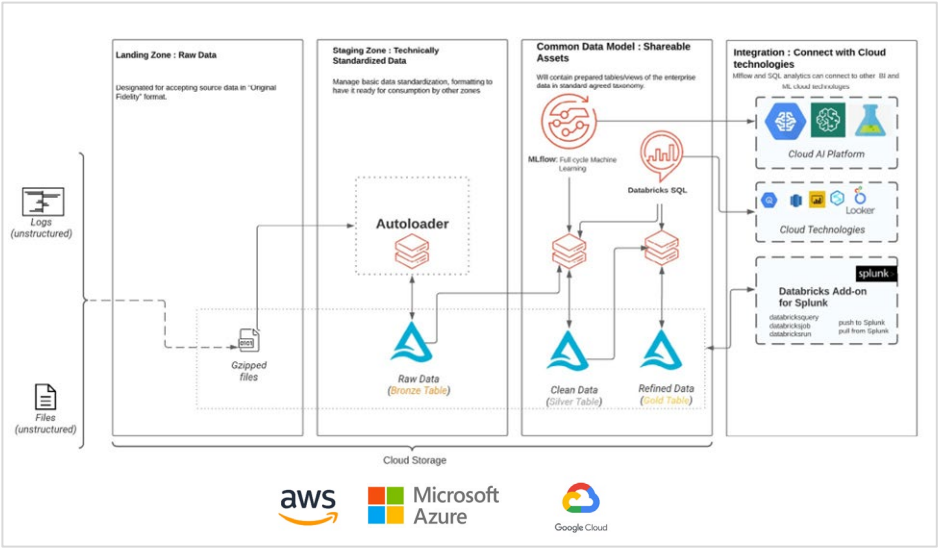


図1： CrowdStrike Falconデータのためのレイクハウスアーキテクチャ

ブロンズからシルバーの段階になると、データに構造を持たせるためのスキーマが追加されます。単一のデータソースから読み取るので、異なるタイプのイベントを全て処理し、それぞれのテーブルに書き込む際に正しいスキーマを強制することができるのです。シルバーレイヤーでスキーマを強制することができるため、ML や分析ワークロードを構築するための堅牢な基盤になります。

ゴールドステージは、ダッシュボードや BI ツールでのクエリや性能の高速化のためにデータを集約するもので、ユースケースやデータ量に応じてオプションで用意されています。予期せぬトレンドが観測された場合、アラートが発動するように設定できます。

また、オプション機能として Splunk のための Databricks アドオンがあり、セキュリティチームは Splunk の快適な環境を離れることなく、Databricks の費用対効果の高いモデルや AI のちからを活用することができるようになります。ユーザーは、アドオンにより、Splunk のダッシュボードや検索バーから Databricks に対してアドホックなクエリを実行できます。ユーザーは、Splunk のダッシュボードから、または Splunk の検索に応答して、Databricks Notebook やジョブを起動することもできます。Databricks との統合は双方向で行われ、ユーザーはノイズの多いデータを要約したり、Databricks で検出を実行して Splunk Enterprise Security に表示させたりできます。ユーザーは、 Databricks Notebook から Splunk の検索を実行することもでき、データの重複を防ぐことができます。

Splunk と Databricks の統合により、日常的に使用しているツールを変更せずに、コストの削減、分析するデータソースの拡張、堅牢な分析エンジンの結果の取得が可能になります。

コードの解説

オートローダはファイルベースのデータ取り込みの最も複雑な部分を抽象化するため、数行のコードで未加工からブロンズレイヤーへの取り込みパイプラインを作成できます。以下は、Delta 取り込みパイプラインの Scala コード例です。CrowdStrike Falcon のイベントレコードの共通のフィールド名は `event_simpleName` です。

```
val crowdstrikeStream = spark.readStream
  .format("cloudFiles")
  .option("cloudFiles.format", "text")  // text file doesn't need schema
  .option("cloudFiles.region", "us-west-2")
  .option("cloudFiles.useNotifications", "true")
  .load(rawDataSource)
  .withColumn("load_timestamp", current_timestamp())
  .withColumn("load_date", to_date($"load_timestamp"))
  .withColumn("eventType", from_json($"value", "struct", Map.empty[String,
String]))
.selectExpr("eventType.event_simpleName","load_date","load_timestamp", "value" )
  .writeStream
  .format("delta")
  .option("checkpointLocation", checkPointLocation)
  .table("demo_bronze.crowdstrike")
```

未加工からブロンズレイヤーでは、未加工データからイベント名のみを抽出します。ロードタイムスタンプと日付のカラムを追加することで、ユーザーはブロンズテーブルに未加工データを保存します。ブロンズテーブルは、イベント名とロード日時でパーティショニングされており、特に限られた数のイベント日時範囲に関心がある場合、ブロンズからシルバーへのジョブをより高い性能で実行するのに役立ちます。

次に、ブロンズからシルバーへのストリーミングジョブがブロンズテーブルからイベントを読み込み、スキーマを強制し、イベント名に基づいて数百のイベントテーブルに書き込みます。以下は、Scala のコード例です。

```
spark
  .readStream
  .option("ignoreChanges", "true")
  .option("maxBytesPerTrigger", "2g")
  .option("maxFilesPerTrigger", "64")
  .format("delta")
  .load(bronzeTableLocation)
  .filter($"event_simpleName" === "event_name")
  .withColumn("event", from_json($"value", schema_of_json(sampleJson)) )
  .select($"event.*", $"load_timestamp", $"load_date")
  .withColumn("silver_timestamp", current_timestamp())
  .writeStream
  .format("delta")
  .outputMode("append")
  .option("mergeSchema", "true")
  .option("checkpointLocation", checkPoint)
  .option("path", tableLocation)
  .start()
```

各イベントスキーマは、スキーマレジストリに保存することも、データドリブンサービス間でスキーマを共有する場合には、Delta テーブルへの保存も可能です。上記のコードでは、ブロンズテーブルから読み込んだサンプルの JSON 文字列を使用し、スキーマは schema_of_json() を使用して JSON から推論されます。その後、from_json() を用いて JSON 文字列を構造体に変換します。その後、構造がフラットになり、タイムスタンプのカラムが追加されました。これらの手順は、イベントテーブルに追加するために必要な全ての列を持つデータフレームを提供します。最後に、構造化されたデータをアペンドモードでイベントテーブルに書き込みます。

また、マイクロバッチを処理する関数を定義することで、foreachBatch で 1 つのストリームで複数のテーブルにイベントをファンアウトさせることも可能です。foreachBatch() を使えば、既存のバッチデータソースを再利用して、複数のテーブルへのフィルタリングや書き込みを行うことが可能です。しかし、foreachBatch() では、最低 1 回の書き込みしか保証されません。そのため、完全一致のセマンティクスを強制するためには、手動による実装が必要です。

この段階で、Databricks Notebook やジョブでサポートされている Python、R、Scala、SQL 言語を使って、構造化データを照会できます。シルバーレイヤーデータは、ML やサイバー攻撃の解析に利用するのに便利です。

次のストリーミングパイプラインは、シルバーからゴールドです。この段階では、ダッシュボードやアラートのためのデータ集計が可能です。このブログシリーズの第 2 回目では、Databricks SQL を使ったダッシュボードの構築方法について、もう少し詳しく説明します。

次のステップ

ユースケースに ML を適用し、Databricks SQL を使用することでさらに価値を高めるブログを今後もお届けします。

これらの Notebook は、お客様の Databricks デプロイメント環境で使用できます。Notebook の各項目にはコメントがついています。 この Notebook をより使いやすくするためのご質問やご意見をお待ちしています。以下のメールアドレス宛にご連絡ください。cybersecurity@databricks.com

Databricks Notebook の無料お試しをご利用ください。

SECTION 2.3

最新のデータレイクハウスで ヘルスケアデータの可能性を最大化

投稿者：Michael Ortega、Michael Sanky、Amir Kermany

2021年7月19日

医療・ライフサイエンス業界におけるデータウェアハウスおよびデータレイクの課題を克服するには

患者1人に対して年間およそ80 MBの医療データが生成されるといわれています。数千人規模の患者の生涯に換算すると、貴重な知見の源となるペタバイト級の患者データが生成されることになります。膨大なデータから知見を抽出することで、臨床業務の効率化、創薬研究の加速、患者の転帰の改善が図れます。これを可能にするためには、データを収集するだけでなく、準備段階として、データの前処理が必要です。ダウンストリームの分析やAIで利用できるよう、収集したデータのクリーニングや構造化を行います。ヘルスケア・ライフサイエンス組織のほとんどが、この準備段階に多大な時間を消費しているのが実情です。

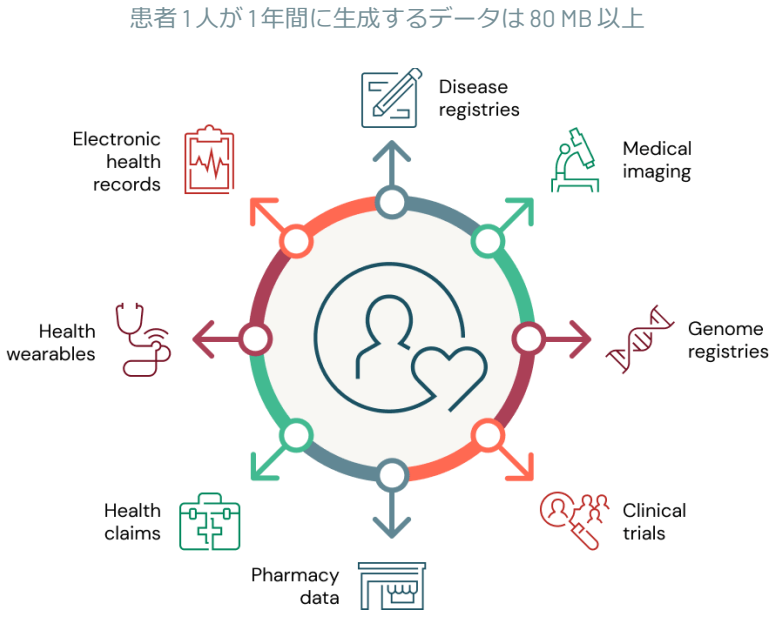


図1：ヘルスケアデータは爆発的に増大しており、患者1人が1年間に生成するデータは80 MBを超える

業界におけるデータ分析の課題

ヘルスケア・ライフサイエンス業界の組織が抱える、データ準備、分析、AI における課題には数多くの理由が存在しますが、そのほとんどは、データウェアハウス（DWH）上に構築されたレガシーなデータアーキテクチャへの投資に関係するものです。この業界における 4 つの主要な課題は次のとおりです。

課題 1： ボリューム
急増するデータに対応するスケーリング

ヘルスケア業界におけるデータの爆発的な増加を示す最適な例は、ゲノミクスです。当初、ゲノムのシーケンスにかかるコストは 10 億ドルを超えていました。膨大なコストゆえ、初期は（今もなお多く）、人間のゲノムのごく一部、通常0.1%程度において特定の変異体を見つけ出すプロセスである、ジェノタイピングに注力されていました。その後、ゲノム上のタンパク質をコードする領域を解析対象とする全エクソームシーケンスへと移行しましたが、それでもゲノム全体の 2% 未満でした。現在、企業は全ゲノムシーケンス（WGS）による顧客への直接検査を提供しており、30 回の WGS で 300 ドル以下となっています。集団レベルにおいては、2021 年に、UK バイオバンクが研究用として 20 万以上の全ゲノムを公開しています。ゲノミクスに限らず、画像、ウェアラブル医療機器、電子医療記録も急速に増大しています。

集団医学分析や創薬などの取り組みへの鍵は、スケーリングです。レガシーアーキテクチャの多くは、オンプレミスに構築されており、ピーク時の容量にあわせて設計されています。そのため、使用率が低い期間はコンピューティングパワーの未使用が発生し（最終的にはコストの無駄）、アップグレードが必要な場合も、迅速にスケーリングできません。

課題 2：バラエティ
多様なヘルスケアデータの分析

ヘルスケア・ライフサイエンスの組織は、それぞれ違いのある膨大な種類のデータを取り扱っています。医療データの 80% 以上が非構造化データであることは広く知られていますが、多くの組織はいまだ構造化データ、従来の SQL ベースの分析向けに設計されたデータウェアハウスに注目しています。非構造化データには、腫瘍学、免疫学、神経学などの分野（コストが増大している分野）において、疾患の診断や進行度の測定に不可欠な画像データや、患者の既往歴や生活歴の完全な把握に欠かせない医療記録のナラティブなテキストデータが含まれます。これらのデータタイプを無視したり後回しにしたりすることはできません。

問題をさらに複雑にしているのは、ヘルスケアのエコシステムが相互に関連しあうようになってきており、ステークホルダーは新たなデータタイプに取り組む必要があることです。例えば、医療提供者は、リスクシェアリング契約の管理と裁定のために請求データを必要とし、保険者は事前承認などのプロセスのサポートや品質評価の推進のために臨床データを必要とするいったことです。これらの組織では多くの場合、新しいデータタイプをサポートするデータアーキテクチャやプラットフォームが欠如しています。

一部の組織では、非構造化データや高度な分析のサポートを目的としてデータレイクに投資していますが、これは新たな課題を引き起こしています。この環境においては、データチームはデータウェアハウスとデータレイクの2つのシステムを管理する必要があり、サイロ化されたツール間でのデータコピーはデータ品質と管理上の問題を引き起こします。

課題 3：速度
リアルタイムな知見を取得するためのストリーミングデータ処理

ヘルスケアは、多くのケースにおいて、生死に関わる問題です。状況は劇的に変化しうるため、日次ベースのバッチデータ処理では不十分なことがほとんどです。介入性治療の成功には、最新情報へのアクセスが不可欠です。病院や国際医療システムでは、命を救うために、敗血症の予測やICU病床のリアルタイム需要予測など、あらゆることにストリーミングデータを使用しています。

さらに、データ処理速度は、ヘルスケアにおけるデジタル革命の主要コンポーネントです。個人はかつてないほど多くの情報にアクセスでき、リアルタイムで治療に影響を及ぼすことができます。例えば、Livongo が提供する持続血糖モニターなどのウェアラブルデバイスは、モバイルアプリにリアルタイムでデータをストリーミングし、パーソナライズされた行動を提案します。

このような早期の成功にもかかわらず、いまだ多くの組織はストリーミングデータの速度に対応可能なデータアーキテクチャを設計していません。信頼性の問題やリアルタイムデータと過去のデータの統合に関する課題が、イノベーションを阻害しています。

課題 4：正確さ
ヘルスケアデータと AI の信頼性

ヘルスケアにおいては、臨床および規制基準により、データの高精度が求められます。ヘルスケアの組織は、公衆衛生上の高いコンプライアンス要件を満たさなければなりません。組織内でのデータの民主化には、ガバナンスが不可欠です。

さらに、人工知能（AI）や機械学習（ML）を臨床現場に導入する場合には、優れたモデルガバナンスが必要となります。しかし、組織の多くは、データサイエンスのワークフロー用にデータウェアハウスと断絶された別のプラットフォームを使用しています。これは、AI を活用したアプリケーションにおけるデータの信頼性と再現性に大きな問題を引き起こします。

レイクハウスでヘルスケアデータの可能性を最大化

レイクハウスアーキテクチャは、クラウドデータレイクの低コスト、スケーラビリティ、柔軟性と、データウェアハウスの性能、ガバナンスを兼ねそろえたモダンデータアーキテクチャであり、ヘルスケア・ライフサイエンスの組織が信頼性と再現性の課題を解決することを支援します。組織では、レイクハウスを用いて、あらゆるタイプのデータを格納でき、オープンな環境で、あらゆる分析、機械学習を実行できます。

ヘルスケア・ライフサイエンス向けレイクハウスの構築

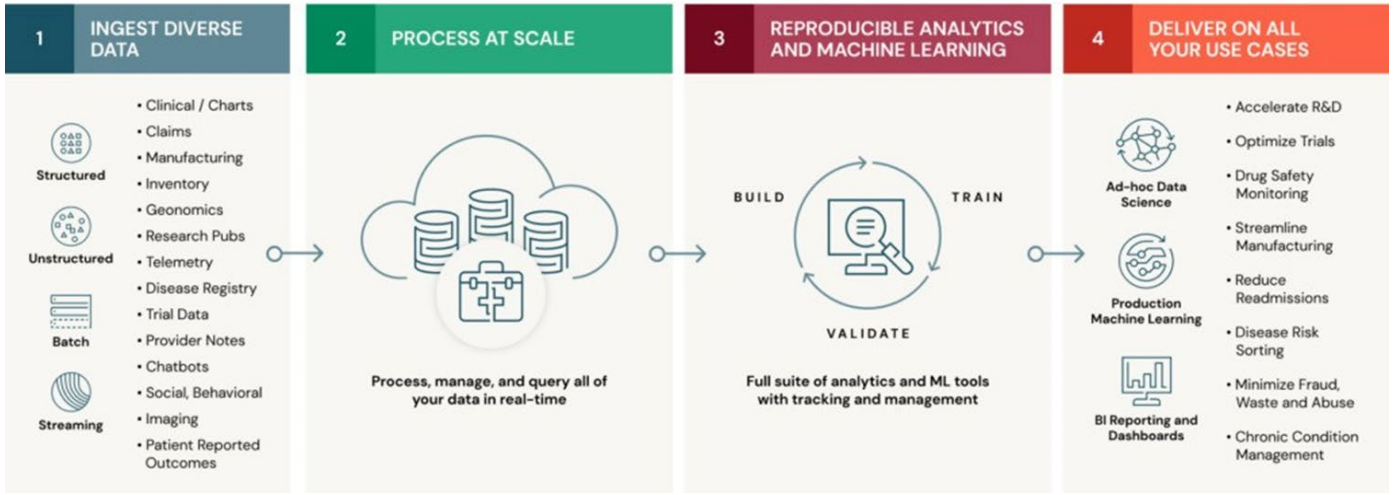


図 2：モダンレイクハウスアーキテクチャにより、ヘルスケア・ライフサイエンスのあらゆるデータ分析のユースケースに対応

レイクハウスは、ヘルスケア・ライフサイエンス組織に次のような利点をもたらします。

- **あらゆるヘルスケアデータを大規模に整理：**Databricks のレイクハウスプラットフォームの中核となる Delta Lake は、オープンソースのデータ管理レイヤーであり、データレイクに信頼性と性能をもたらします。Delta Lakeは従来型のデータウェアハウスとは異なり、あらゆる構造化・非構造化データをサポートします。Databricks は、EMR やゲノミクスなどの分野固有のデータタイプに対応したコネクタを開発し、ヘルスケアデータの取り込みを容易にしています。これらのコネクタは、すぐに使えるソリューションアクセラレータに業界標準のデータモデルがパッケージされています。さらに、Delta Lake には、データのキャッシングやインデックス作成を可能にする最適化機能を備えており、データの処理速度は大幅に向上しています。

これらの機能により、チームは全ての未加工データを一元管理でき、患者の健康状態に関する全体像を把握できます。

- **患者分析と AI を最大活用：**レイクハウスにデータが一元化されることで、チームは、データに対して直接、患者分析の実施、予測モデルの構築ができるようになります。Databricks は、分析および AI ツールを備え、SQL、R、Python、Scala などの幅広いプログラミング言語をサポートするコラボレーティブなワークスペースを提供し、このようなケイパビリティを可能にします。データサイエンティスト、データエンジニア、医療インフォマティシストなど、さまざまなユーザーグループがコラボレーションをしながら、ヘルスケアデータの分析、モデル構築、可視化を行うことができます。

- **患者に関する知見をリアルタイムで提供：**レイクハウスは、ストリーミング、バッチデータに対する統合アーキテクチャを提供します。2 つの異なるアーキテクチャを管理する必要はなく、信頼性の問題で頭を抱えることもありません。さらに、Databricks 上でレイクハウスアーキテクチャを運用することで、企業はワークロードに応じて自動的にスケールする、クラウドネイティブなプラットフォームにアクセスできるようになります。これにより、ストリーミングデータの取り込み、ペタバイト規模の履歴データとの組み合わせが容易になり、集団規模でのニアリアルタイムの知見を抽出できます。
- **データ品質とコンプライアンスを提供：**レイクハウスには、スキーマ強制、監査、バージョン管理、高粒度のアクセス制御など、従来のデータレイクにはない機能が備わっており、データの信頼性を高めています。レイクハウスによる重要なメリットは、同一で信頼性の高いデータソースに、分析と機械学習の両方を実行できることです。加えて、Databricks は機械学習モデルのトラッキングや管理機能を提供し、チームが環境間で結果を再現することを容易にしており、コンプライアンス基準を満たすサポートをしています。これらの機能は全て、HIPAA 準拠の分析環境で提供されます。

レイクハウスは、ヘルスケア・ライフサイエンスデータの管理において、最適なアーキテクチャです。このアーキテクチャと Databricks の機能の相乗効果により、組織は、創薬や慢性疾患管理プログラムなど、極めてインパクトのある幅広いユースケースへの対応を支援できるようになります。

ヘルスケア・ライフサイエンスに特化したレイクハウスの構築を始めるには

Databricks では、ヘルスケア・ライフサイエンスの組織が特定のニーズにあわせてレイクハウスの構築ができるように、一連のソリューションアクセラレータを提供しています。ソリューションアクセラレータには、Databricks Notebook にあるサンプルデータ、事前構築されたコード、ステップごとの説明が含まれています。

新しいソリューションアクセラレータ：リアルワールドエビデンスのためのレイクハウス

リアルワールドデータは、製薬会社に臨床試験外の患者の健康状態や薬効に関する新たな知見を提供します。このアクセラレータでは、Databricks 上にリアルワールドエビデンスのレイクハウスを構築できます。患者集団に対するサンプルの EHR データの取り込みから、OMOP 共通データモデルを用いたデータの構造化、薬剤の処方パターンの調査まで、大規模な分析を実行する方法を紹介します。

ヘルスケア・ライフサイエンス業界向けソリューションについて詳しくは、こちらをご覧ください。

Databricks Notebook の無料お試しをご利用ください。

SECTION 2.4

規制関連のレポーティングの即時性と信頼性の確保

投稿者：Antoine Amend、 Fahmid Kabir

2021年9月17日

リスクおよび規制に対するコンプライアンスの管理は、いっそう複雑かつコストのかかる取り組みとなっています。2008 年のグローバル金融危機以来、規制の変更は 500% 増加し、規制に対応するためのコストを引き上げています。規制に準拠しないこと、SLA 違反による罰金（2019年の AML における罰金は 100 億ドルにもなっており、銀行は常に高い罰金を科せられています）のため、データが不完全な状態でもレポートの処理は行われなくてはなりません。一方、貧弱なデータ品質の追跡レコードに対しては「管理不十分」であることから「罰金」が科せられます。結果として、多くの金融サービス機関（FSI）が、データの信頼性と即時性のバランスをとることで、貧弱なデータ品質と厳格な SLA と戦い続けることになります。

この規制レポートに関するソリューションアクセラレータでは、規制の SLA に対応するために、Delta Live Tables（DLT）がどのようにリアルタイムで規制データを獲得し、処理を保証するのかを示します。Delta Sharing と Delta Live Tables を組み合わせることで、アナリストは、リアルタイムで信頼性のある高品質な規制データを得ることができます。このブログ記事では、金融サービス業界のデータモデルとクラウドコンピューティングの柔軟性を組み合わせ、少ない開発オーバヘッドでガバナンス標準を実現するレイクハウスアーキテクチャの利点をご紹介します。FIRE データモデルとは何か、堅牢なデータパイプラインを構築するためにどのように DLT が統合されるのかを説明します。

FIRE データモデル

Financial Regulatory data standard（FIRE）は、金融業界における規制システム間でデータを送信する際の一般的な仕様を定義しています。規制データとは、規制当局への提出物、要件、計算の基礎となるデータを指し、政策、監視、監督の目的で使用されます。FIRE データ標準は、Horizon 2020 資金提供プログラムを通じて、欧州委員会、オープンデータ研究所、および、欧州オープンデータインキュベータ の FIRE データ標準によってサポートされています。このソリューションの一部として、Databricks は、FIRE データモデルを Apache Spark™ 運用パイプラインに解釈できる PySpark モジュールを提供しました。

Delta Live Tables

Databricks は先日、データパイプラインのオーケストレーションに関する新製品「Delta Live Tables」を発表しました。Delta Live Tables を用いることで、企業では信頼性のあるデータパイプラインの構築、管理が容易になります。複数のエクスペクテーションを管理し、リアルタイムで不正なレコードを削除あるいはモニタリングすることができるため、Delta Live Tables と FIRE データモデルを統合することのメリットは明らかです。以下のアーキテクチャに示すように、Delta Live Tables はクラウドストレージに到着する詳細な規制データを取り込み（**ingest**）、コンテンツにスキーマを適用（**schematize**）し、一貫性を保つために FIRE データ仕様に従ってレコードを検証（**validate**）します。Delta Sharing を使用して、安全、スケーラブル、透明性のある方法で規制システム間で細粒度の情報を交換するデモを見ていきましょう。

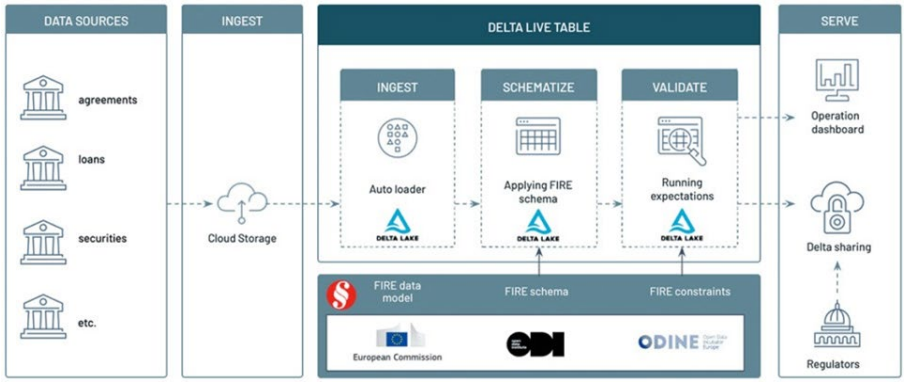


図 1

スキーマの適用

JSON ファイルなど、一部のデータフォーマットは構造化されているように見えても、スキーマの適用は必要です。エンタープライズ環境、特に規制コンプライアンス下では、スキーマの適用は、期待するフィールドの除外、フィールドの削除、データタイプの完全な評価（例：日付は文字列でなく date オブジェクトとして扱う）を保証します。また、データドリフトに対するシステムの耐性試験にもなります。FIRE PySpark モジュールを用いることで、未加工のレコードのストリームに適用される指定の FIRE エンティティ（この例では collateral エンティティ）を処理するのに必要な Spark スキーマをプログラムから取得できます。

```
from fire.spark import FireModel
fire_model =
FireModel().load("
collateral")
fire_schema =
fire_model.schema
```

以下の例では、到着する CSV ファイルにスキーマを適用しています。@dlt アノテーションでプロセスを修飾することで、Delta Live Tables に対するエントリーポイントを定義し、未加工の CSV ファイルをマウントされたディレクトリから読み込み、スキーマが適用されたレコードをブロンズレイヤーに書き込みます。

```
@dlt.create_table()
def collateral_bronze():
  return (
    spark
      .readStream
      .option("maxFilesPerTrigger", "1")
      .option("badRecordsPath", "/path/to/invalid/collateral")
      .format("csv")
      .schema(fire_schema)
      .load("/path/to/raw/collateral")
```

エクスペクテーションの評価

スキーマの適用とは別に、制約を適用することができます。FIRE エンティティのスキーマ定義（collateral のスキーマ定義例をご覧ください）に基づいて、フィールドが必要か不要かを検知できます。列挙オブジェクト（例：国コード）を指定することで、値の一貫性を保証することができます。スキーマによる技術的制約に加え、FIRE モデルでは最小値、最大値、金額、最大個数などビジネス上の制約を定義しています。プログラムがこれら全ての技術的、ビジネス上の制約を FIRE データモデルから取得し、一連の Spark SQL エクスペクテーションとして解釈します。

```
from fire.spark import FireModel
fire_model = FireModel().load("collateral")
fire_constraints = fire_model.constraints
```

Delta Live Tables を用いることで、ユーザーは複数のエクスペクテーションを一度に評価して不正なレコードを削除でき、データ品質をモニタリングしたり、パイプライン全体を停止できます。私たちのシナリオでは、この記事で提供されている Notebook にあるように、エクスペクテーションに違反したレコードは削除したいと考え、削除レコードは検疫テーブルに格納しました。

```
@dlt.create_table()
@dlt.expect_all_or_drop(fire_constraints)
def collateral_silver():
  return dlt.read_stream("collateral_bronze")
```

数行のコードで、シルバーテーブルが文法的（適切なスキーマ）にも、意味的（適切なエクスペクテーション）にも正しいことを保証できます。以下に示すように、コンプライアンスオフィサーはリアルタイムで処理されているレコード件数に対する完全な可視性を得ることができます。この例では、collateral エンティティは 92.2% 処理されているのを確認しました（残りの 7.8% は検疫で処理されます）。

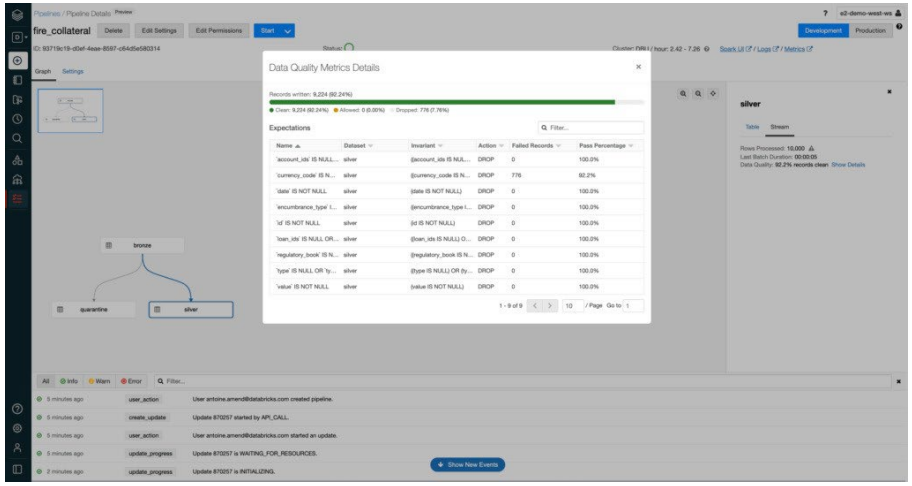


図 2

オペレーションデータストア

Delta Live Tables は実際のデータを Delta ファイルに格納することに加え、オペレーションメトリクスを Delta フォーマットで system/events 配下に格納します。ここでは、レイクハウスの標準的なパターンに従い、オートローダを用いて新規オペレーションメトリクスを「サブスクライブ」し、バッチあるいはリアルタイムでシステムメトリクスを処理します。あらゆるデータの更新を追跡する Delta Lake のトランザクションログを活用することで、企業はチェックポイントプロセスを構築、管理する必要なしに新規メトリクスにアクセスできます。

```
input_stream = spark \
  .readStream \
  .format("delta") \
  .load("/path/to/pipeline/system/events")

output_stream = extract_metrics(input_stream)

output_stream \
  .writeStream \
  .format("delta") \
  .option("checkpointLocation", "/path/to/checkpoint") \
  .table(metrics_table)
```

	entity	expectation_name	expectation_value
1	adjustment	[id] is mandatory	`id` IS NOT NULL
2	adjustment	[date] is mandatory	`date` IS NOT NULL
3	adjustment	[col] is mandatory	`col` IS NOT NULL
4	adjustment	[contribution_amount] is mandatory	`contribution_amount` IS NOT NULL
5	adjustment	[currency_code] is mandatory	`currency_code` IS NOT NULL
6	adjustment	[currency_code] not allowed value	(`currency_code` IS NULL) OR (`currency_code` IN ('AED', 'AFN', 'ALL', 'AMD', 'ANG', 'AOA', 'ARS', 'AUD', 'AWG', 'AZN', 'BAM', 'BBD', 'BDT', 'BGN', 'BHD', 'BIF', 'BMD', 'BND', 'BOB', 'BOV', 'BRL', 'BSD', 'BTN', 'BWP', 'BYN', 'BZD', 'CAD', 'CDF', 'CHE', 'CHF', 'CHW', 'CLF', 'CLP', 'CNY', 'COP', 'COU', 'CRC', 'CUC', 'CUP', 'CVE', 'CZK', 'DJF', 'DKK', 'DOP', 'DZD', 'EGP', 'ERN', 'ETB', 'EUR', 'FJD', 'FKP', 'GBP', 'GEL', 'GHS', 'GIP', 'GMD', 'GNF', 'GTQ', 'GYD', 'HKD', 'HNL', 'HRK', 'HTG', 'HUF', 'I...

図 3

全てのメトリクスがオペレーションストアで集中管理されることで、アナリストは Databricks SQL を活用して、シンプルなダッシュボードや、リアルタイムでデータ品質問題を検知するためにより複雑なアラート機構を構築することができます。

Delta Live Tables によって提供されるデータ品質の透明性と、Delta Lake フォーマットの不変性の特性を組み合わせることで、金融機関は、規制コンプライアンスに求められるデータボリューム、データ品質の両方にマッチする特定バージョンに「タイムトラベル」することが可能となります。この例では、検疫に格納された7.2%の不正レコードのリプレイはシルバーテーブルにアタッチされた別の Delta バージョンとなり、このバージョンは規制機関と共有できます。

```
DESCRIBE HISTORY fire.collateral_silver
```

規制データの送信

データの品質とボリュームに対して信頼性を得ることができ、金融機関は企業データ交換に対するオープンプロトコルである Delta Sharing を使用して、規制システム間でセキュアな情報交換ができるようになります。ユーザーが同じプラットフォームを使用することや、データを消費するために複雑なETLパイプライン（例えば、SFTP サーバー経由でのデータファイルへのアクセス）に依存することなしに、Delta Lake の持つオープンソースの特性によって、データの消費者は Python、Spark、あるいは MI／BI ダッシュボード（Tableau、PowerBI など）から、スキーマが適用されたデータに直接アクセスできます。

シルバーテーブルをそのままの状態で共有することもできますが、データ品質が事前に定義された基準を達成しているときのみに規制データを共有したいと考えるかもしれません。この例では、シルバーテーブルを別のバージョンとして、内部ネットワークとは隔離された別の場所（非武装地帯：DMZ）にコピーしてエンドユーザーによってアクセスできるようにします。

```
from delta.tables import *

deltaTable = DeltaTable.forName(spark, "fire.collateral_silver")
deltaTable.cloneAtVersion(
  approved_version,
  dmz_path,
  isShallow=False,
  replace=True
)

spark.sql(
  "CREATE TABLE fire.colleral_gold USING DELTA LOCATION '{}'"
    .format(dmz_path)
)
```

Delta Sharing オープンソースソリューションはアクセス権を管理するために共有サーバーを利用しますが、Databricks では Unity Catalog を活用して、アクセスコントロールポリシーを強制、集中管理して完全な監査ログ機能を提供し、SQL インタフェースを通じてアクセス管理をシンプルにします。以下の例では、規制テーブルを含む SHARE と、データを共有する RECIPIENT を作成しています。

```
-- DEFINE OUR SHARING STRATEGY
CREATE SHARE regulatory_reports;

ALTER SHARE regulatory_reports ADD TABLE fire.collateral_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.loan_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.security_gold;
ALTER SHARE regulatory_reports ADD TABLE fire.derivative_gold;

-- CREATE RECIPIENTS AND GRANT SELECT ACCESS
CREATE RECIPIENT regulatory_body;

GRANT SELECT ON SHARE regulatory_reports TO RECIPIENT regulatory_body;
```

アクセスが許可された規制機関、ユーザーは、このプロセスを通じて交換されるパーソナルアクセストークンを用いて背後のデータにアクセスできます。Delta Sharing の詳細に関しては、製品ページを参照いただくか、Databricks 担当にお問い合わせください。

コンプライアンスに対する証明テスト

一連の Notebook および Delta Live Tables のジョブを通じて、規制データの取り込み、処理、検証、送信におけるレイクハウスアーキテクチャのメリットをご紹介しました。特に、一般的なデータモデル（FIRE）と柔軟性のあるオーケストレーションエンジン（Delta Live Tables）を組み合わせることで、規制パイプラインの一貫性、完全性、即時性を容易に実現できることを解説しました。Delta Sharing 機能により、FSI が報告要件を満たし、運用コストを削減し、新しい基準に適応しながら、さまざまな規制システム間で交換される規制データに完全な透明性と信頼性をもたらす方法を最終的に実証しました。

この Notebook は FIRE データパイプラインの理解に役立ちます。金融サービス向け最新のソリューションは Databricks ソリューションアクセラレータの一覧 をご覧ください。

Databricks Notebook の無料お試しをご利用ください。

SECTION 2.5

Databricks レイクハウスプラットフォームを利用した AML ソリューション

投稿者：Sri Ghattamaneni、 Ricardo Portilla、 Anindita Mahapatra
2021年7月16日

金融犯罪ソリューション構築のための重要な課題を解決する

アンチマネーロンダリング（AML）コンプライアンスは、世界中の金融機関を監督する規制当局にとって、間違いなく最重要課題の1つとなっています。数十年にわたる AML の進化と高度化に伴い、現代のマネーロンダリングやテロ資金調達のスキームに対抗するための規制要件も変化しています。1970 年に制定された銀行機密保護法は、金融機関が金融取引を監視し、疑わしい財務活動を関係当局に報告するために適切な管理を行うための指針と枠組みです。この法律により、金融機関がマネーロンダリングや金融テロに対抗するための枠組みが整いました。

アンチマネーロンダリングが複雑な理由

現在の AML の運用は、過去 10 年間の運用とは全く異なっています。金融機関（FI）はデジタルバンキングに移行し、1日あたり数十億規模の取引を処理しています。厳格な取引監視システムや堅牢な Know Your Customer（KYC）ソリューションを使っていたとしても、マネーロンダリングは範囲をますます拡大しています。このブログでは、金融機関のお客様と協力して、レイクハウスプラットフォーム上に企業規模の AML ソリューションを構築し、強力な監視を提供するとともに、現代のオンラインマネーローンダリングの脅威に対応する革新的で拡張性の高いソリューションを実現した経験をご紹介します。

レイクハウスによる AML ソリューションの構築

1日に何十億もの取引を処理するための運用負担は、複数のソースからのデータを保存し、集約的な次世代 AML ソリューションを動かす必要性から生じています。これらのソリューションは、強力なリスク分析とレポーティングを提供し、高度な機械学習モデルの使用をサポートして、誤検知を減らし、下流の調査効率を向上させます。金融機関は既に、より優れたセキュリティと俊敏性、そして膨大な量のデータを保存するために必要な規模の経済を求めて、オンプレミスからクラウドに移行し、インフラとスケーリングの問題を解決するための手段を講じています。

しかし、安価なオブジェクトストレージに収集、保存された膨大な量の構造化、非構造化データをどのように意味づけるかという問題があります。クラウドベンダーはデータを保存するための安価な方法を提供しますが、ダウンストリームの AML リスク管理およびコンプライアンス活動のためにデータを活用するには、ダウンストリームで利用できる高品質・高性能な形式でデータを保存することが必要です。Databricks のレイクハウスプラットフォームは、これを実現します。データレイクの低ストレージコストとデータウェアハウスがもたらす堅牢なトランザクション機能を組み合わせることで、金融機関は最新の AML プラットフォームを構築できます。

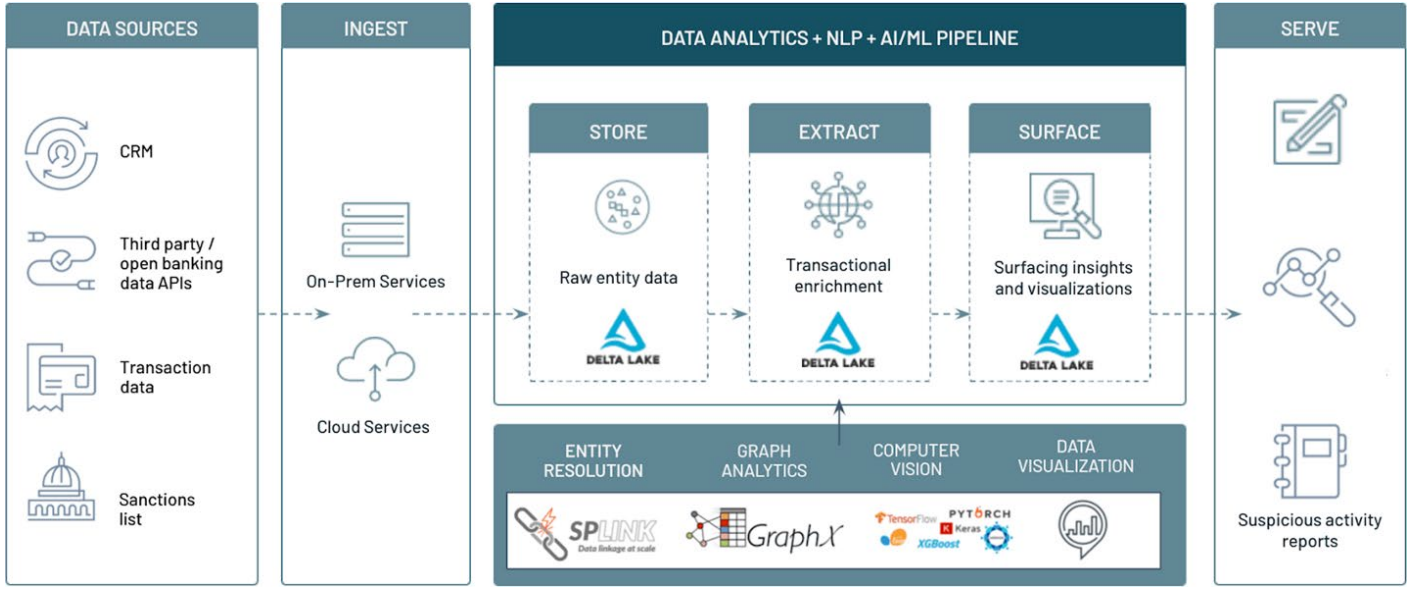


図 1

AML アナリストは、このようなデータストレージの課題に加え、次のようなドメイン特有の重要な課題にも直面しています。

- **非構造化データの処理：**画像、テキストデータ、ネットワークリンクなどの非構造化データの解析にかかる時間短縮の実現
- **DevOps の負担軽減：**エンティティの解決、コンピュータビジョン、エンティティメタデータのグラフ分析など、重要なML機能をサポートする DevOps の負担の軽減
- **サイロの破壊：**AML トランザクションとエンリッチドテーブルに分析エンジニアリングとダッシュボードレイヤーを導入してサイロを破壊する

Databricks は Delta Lake を活用して非構造化データと構造化データの両方を保存、結合し、エンティティリレーションシップを構築することでこれらの解決を支援します。さらに、Databricks の Delta エンジンは新しい Photon コンピュートを使って効率的なアクセスを提供し、テーブルに対する BI クエリを高速化します。これらの機能に加えて、機械学習はレイクハウスの一級市民です。つまり、アナリストやデータサイエンティストは、ダッシュボードを共有し、悪質な業者の一歩先を行くために、サブサンプリングやデータの移動に時間を浪費せずにすみます。

グラフ機能による AML パターンの検知

AML アナリストが案件の一部として使用する主なデータソースの 1 つは、トランザクションデータです。このデータは表形式で、SQL で簡単にアクセスできるにもかかわらず、SQL クエリで 3 層以上の深いトランザクションのチェーンを追跡することが困難です。そのため、不審な個人がつながったネットワークで違法な取引を行うといった単純な概念を表現する柔軟な言語群や API が重要になります。機械学習向け Databricks ランタイムは、グラフ API 「GraphFrames」が事前インストールされており、追跡を容易にします。

このセクションでは、グラフ分析を使って、合成 ID やレイヤリング／構造化などの AML スキームを検出する方法を紹介します。トランザクションとトランザクションから派生したエンティティからなるデータセットを活用し、Apache Spark™、GraphFrames、Delta Lake を用いてこれらのパターンの存在を検知します。Databricks SQL は、これらの知見を集約したゴールドレベルのものに対して適用され、エンドユーザーにグラフ解析のパワーを提供します。

シナリオ 1：合成 ID

合成 ID の存在は警戒すべきものです。グラフ解析を使えば、取引先のエンティティを一括して分析し、リスクレベルを検知できます。私たちの分析では、これを 3 つのフェーズで実行します。

- トランザクションデータに基づき、エンティティを抽出する
- 住所、電話番号、電子メールに基づくエンティティ間のリンクを作成する
- GraphFrames の接続コンポーネントを使用して、複数のエンティティ（上記の IDおよびその他の属性で識別）が 1 つまたは複数のリンクで接続されているかどうかを判断する

エンティティ間に存在するコネクション（＝共通属性）の数に基づいて、リスクスコアを低くしたり高くしたりして、スコアの高いグループに基づくアラートを作成できます。以下は、この考え方の基本を表しています。

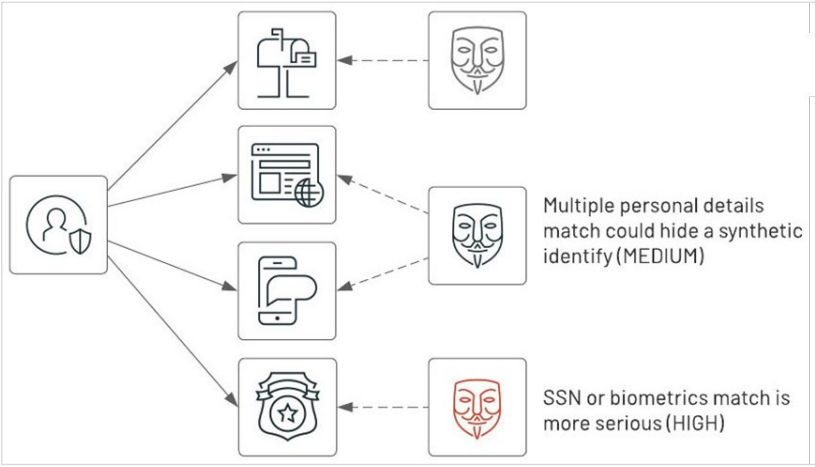


図 2

まず、住所、メールアドレス、電話番号を使って ID グラフを作成し、これらの属性のいずれかに一致する個人を結びつけます。

```
e_identity_sql = '''
select entity_id as src, address as dst from  aml.aml_entities_synth where
address is not null
UNION
select entity_id as src, email as dst from aml.aml_entities_synth where
email_addr is not null
UNION
select entity_id as src, phone as dst from aml.aml_entities_synth where
phone_number is not null
'''

from graphframes import *
from pyspark.sql.functions import *
aml_identity_g = GraphFrame(identity_vertices, identity_edges)
result = aml_identity_g.connectedComponents()

result \
  .select("id", "component", 'type') \
  .createOrReplaceTempView("components")
```

次に、2 つのエンティティの個人識別情報とスコアが重複している場合に識別するためのクエリを実行します。これらのグラフコンポーネントのクエリの結果から、一致する 1 つの属性（住所など）のクエリなるコホートが予想されますが、これはあまり心配する必要はありません。しかし、一致する属性が増えれば、アラートが出る可能性も高くなります。下図のように、3 つの属性が全て一致するケースにフラグを立てることで、SQL アナリストが全てのエンティティに対して実行したグラフ分析の結果を日次で取得できます。

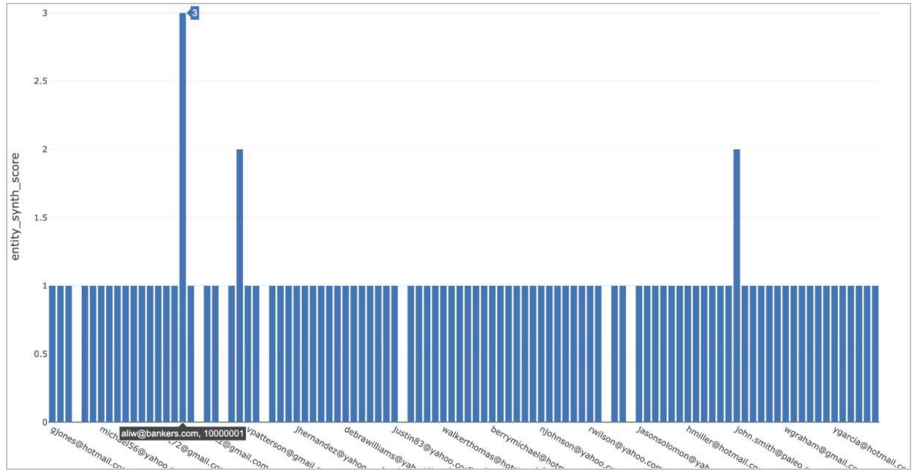


図 3

シナリオ 2：構造化

もう一つのよくあるパターンは、「構造化」と呼ばれるものです。これは、複数の企業が結託し、「目立たないように」小さな支払いを複数の銀行に送り、その後、より大きな総額が最終的な金融機関に送られるというものです（下図右端）。このシナリオでは、全ての関係者が、通常当局に警告が出される1万ドルの基準額を下回っています。グラフ解析で簡単に実現できるだけでなく、モチーフを見つける技術を自動化して、他のネットワークの組み合わせに拡張し、同じ方法で他の疑わしい取引を見つけることが可能です。

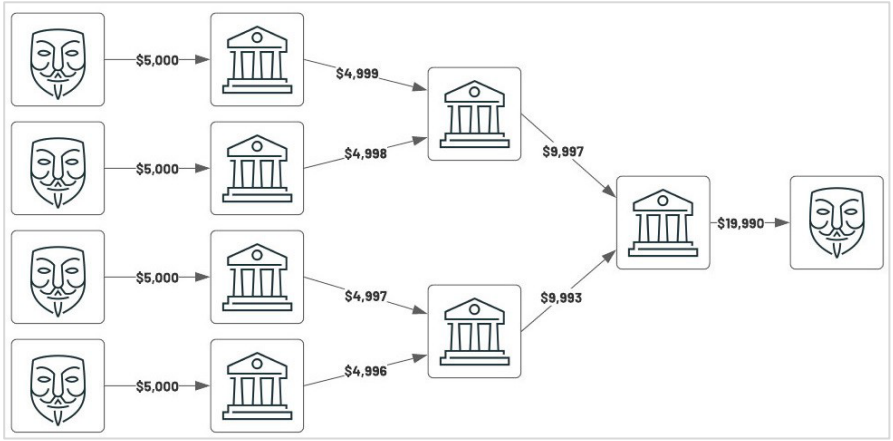


図 4

では、グラフ機能を用いて上記のシナリオを検出するための基本的なモチーフ探索コードを記述します。この出力は半構造化 JSON であることに注意してください。非構造化タイプを含む全てのデータ型は、レイクハウスで簡単にアクセスできます。また、 これらの特定の結果は、SQL レポート用に保存されます。

```
motif = “(a)-[e1]->(b); (b)-[e2]->(c); (c)-[e3]->(d); (e)-[e4]->(f); (f)-
[e5]->(c); (c)-[e6]->(g)”
struct_scn_1 = aml_entity_g.find(motif)

joined_graphs = struct_scn_1.alias(“a”) \
  .join(struct_scn_1.alias(“b”), col(“a.g.id”) == col(“b.g.id”)) \
  .filter(col(“a.e6.txn_amount”) + col(“b.e6.txn_amount”) > 10000)
```

モチーフ発見を利用して、4 つの異なる事業体を経由してお金が流れ、1万ドルのしきい値以下に保たれている興味深いパターンを抽出しました。グラフのメタデータを構造化されたデータセットに結びつけ、AML アナリストがさらに調査できるような分析結果を生成しています。

	top_entity_id	first_entity	second_entity	third_entity	fourth_entity
1	1	Brenda Thomas	Teresa Gibson	Mary Strong	Robert Wilkinson
2	3	Lindsey Barber	Joshua Harris	Mary Strong	Robert Wilkinson
3	5	Bruce White	Kathleen Elliott	Victor Arias	Robert Wilkinson
4	7	Jeffrey Lara	Amy Campbell	Victor Arias	Robert Wilkinson

図 5

シナリオ 3：リスクスコアの伝達

特定された高リスクの事業者は、その輪に影響を与える（ネットワーク効果）ことになります。そのため、影響範囲を反映させるために、相互作用する全てのエンティティのリスクスコアを調整する必要があります。反復的なアプローチを用いることで、任意の深さまでの取引フローを追跡し、ネットワーク内で影響を受ける他の人のリスクスコアを調整できます。前述したように、グラフ解析の実行により、何度も繰り返される SQL 結合や複雑なビジネスロジックを回避でき、メモリの制約により性能に影響を与える可能性があります。グラフ解析と Pregel API は、まさにそのために作られたものです。Google が開発した Pregel は、任意の頂点からその近傍に再帰的にメッセージを「伝播」させ、各ステップで頂点の状態（ここではリスクスコア）を更新することが可能です。Pregel API を用いたダイナミックリスクアプローチは、以下のように表すことができます。

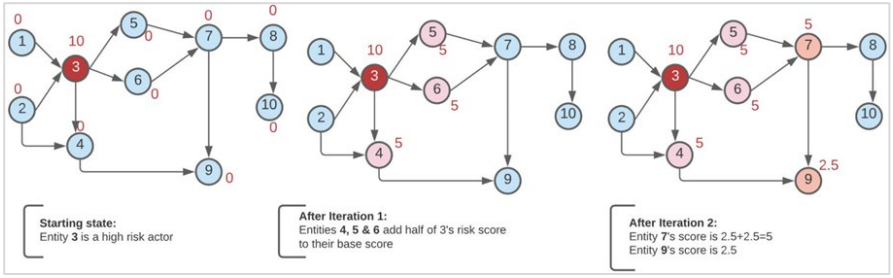


図 6

図 6 は、ネットワークの開始状態と、その後の 2 回の反復を表しています。例えば、リスクスコアが 10 の悪役（ノード 3）を1 人選んでスタートしたとします。そのノード（すなわちノード 4、5、6）と取引して資金を受け取った全ての人に、例えば悪役のリスクスコアの半分を渡して、それを自分の基本スコアに追加することでペナルティを与えたいと思います。次の反復では、ノード 4、5、6 の下流にある全てのノードのスコアが調整されます。

ノード #	反復 #0	反復 #1	反復 #2
1	0	0	0
2	0	0	0
3	10	10	10
4	0	5	5
5	0	5	5
6	0	5	5
7	0	0	5
8	0	0	0
9	0	0	2.5
10	0	0	0

GraphFrame の Pregel API を使うことで、この計算を行い、ダウンストリーム他のアプリケーションが消費できるように修正したスコアを永続化できます。

```
from graphframes.lib import Pregel

ranks = aml_entity_g.pregel \
  .setMaxIter(3) \
  .withVertexColumn(
    "risk_score",
    col("risk"),
    coalesce(Pregel.msg()+ col("risk"),
    col("risk_score"))
  ) \
  .sendMsgToDst(Pregel.src("risk_score")/2 ) \
  .aggMsgs(sum(Pregel.msg())) \
  .run()
```

アドレスマッチング

実際のストリートビュー画像とテキストをマッチングするアドレスマッチングについて簡単に説明します。多くの場合、AML アナリストは、登録されたエンティティにリンクされたアドレス（住所）の正当性を検証します。その住所は商業ビルなのか、住宅地なのか、それとも単なる郵便ポストなのか？写真の解析には、取得、クリーニング、検証など、煩雑な手作業が必要になります。レイクハウスデータアーキテクチャは、Python と PyTorch による ML ランタイムと、事前学習したオープンソースモデルを使用して、このタスクの大部分の自動化を可能にします。右図は人間が見て有効な住所の例です。検証の自動化には、あらかじめトレーニングされた VGG モデルを使用します。このモデルには、住宅の検知に利用できる数百のオブジェクトがあります。

図 7

以下のコードを日次で自動で実行することで、それぞれの画像にラベルを付与できます。また、クエリが容易にできるように、全ての画像参照とラベルを SQL テーブルにロードしています。Delta Lake でこのような非構造化データを照会できることは、アナリストの業務を効率化し、検証プロセスを数日、数週間ではなく、数分に短縮できます。

```
from PIL import Image
from matplotlib import cm

img = Image.fromarray(img)
...

vgg = models.vgg16(pretrained=True)
prediction = vgg(img)
prediction = prediction.data.numpy().argmax()
img_and_labels[i] = labels[prediction]
```

まとめてみると、いくつかの興味深いカテゴリーが現れてくることに気づきます。図 8 の内訳からわかるように、パティオ、モバイルホーム、モータースクーターなど、住宅の住所で検出されたアイテムとして表示されると思われる明らかなラベルがいくつかあります。一方、CV モデルは、1 枚の画像の中で、太陽電池パネルを周囲の物体からラベリングしています。（注：カスタム画像セットで学習していないオープンソースモデルに制限されているため、太陽電池のラベルは正確ではありません。）さらに画像を分析すると、① ここには本物の反射板はなく、② この住所は本物の住宅ではないことがすぐにわかります（図 7 の横並び比較の写真参照）。Delta Lake のフォーマットでは、構造化されていないデータへの参照をラベルとともに保存し、以下のような分類で簡単にクエリを実行できます。

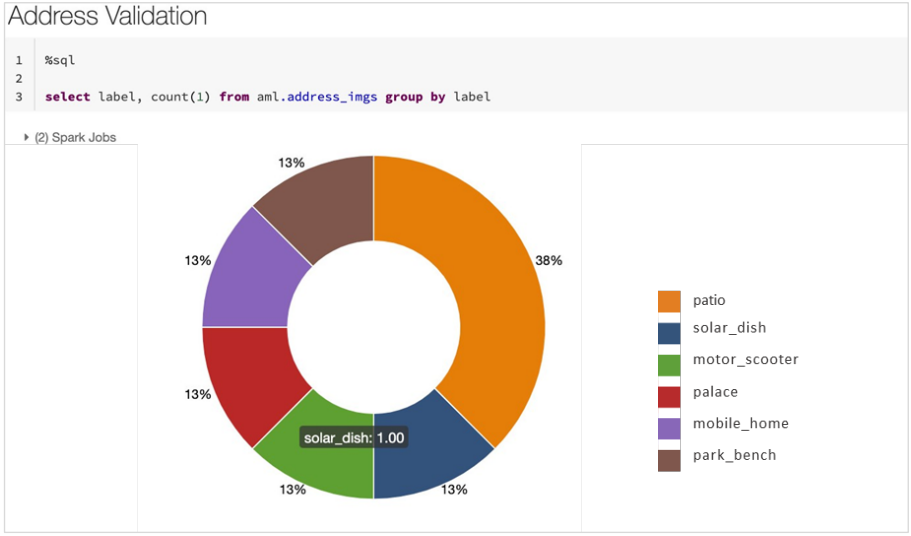


図 8

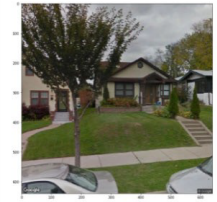
Image Name	Rendered Image	Main Object	Risk Level
img_0.jpg		Patio	Low
img_1.jpg		Solar Dish	High

図 9

エンティティの解決

最後に、AML の課題として注目したいのは、「エンティティの解決」です。多くのオープンソースライブラリがこの問題に取り組んでいるため、いくつかの基本的なエンティティファジーマッチングでは、大規模なリンケージを実現し、一致する列とブロッキングルールを指定する構成を提供する Splink を強調することにしました。

私たちのトランザクションから派生したエンティティのコンテキストでは、Delta Lake のトランザクションを Splink のコンテキストに挿入するのは簡単な作業です。

```
settings = {
   "link_type": "dedupe_only",
   "blocking_rules": [
      "l.txn_amount = r.txn_amount",
   ],
   "comparison_columns": [
      {
         "col_name": "rptd_originator_address",
      },
      {
         "col_name": "rptd_originator_name",
      }
   ]
}
from splink import Splink
linker = Splink(settings, df2, spark)
df2_e = linker.get_scored_comparisons()
```

Splink は、エンティティ属性が極めて類似した取引の特定に使用できる一致確率を割り当てることで、報告された住所、エンティティ名、取引額に関する潜在的な警告を発生させることができます。アカウント情報を照合するエンティティの解決は、極めて手作業が多いという事実を考えると、このタスクを自動化し、情報を Delta Lake に保存するオープンソースのライブラリがあれば、調査官の事件解決の生産性を格段に向上させることができます。エンティティマッチングにはいくつかのオプションがありますが、仕事に適したアルゴリズムを特定するため局所性鋭敏型ハッシュ（LSH）を使用することをお勧めします。LSH とその効果については、ブログで詳しくご紹介しています。

NY Mellon 銀行の住所については、「Canada Square, Canary Wharf, London, United Kingdom」が「Canada Square, Canary Wharf, London, UK」と類似しており、すぐに矛盾が判明しました。重複を排除したレコードを Delta テーブルに戻して保存し、AML 調査に利用できます。

unique_id_l	unique_id_r	rptd_originator_address_l	rptd_originator_address_r
223254	223256	Canada Square, Canary Wharf, London, United Kingdom	Canada Square, Canary Wharf, London, UK

図 10

AML レイクハウスダッシュボード

レイクハウス上の Databricks SQL は、データ管理の簡素化、新しいクエリエンジン Photon による性能、ユーザーの並行処理という点で、従来のデータウェアハウスとのギャップを縮めています。多くの組織では、金融犯罪に対抗するためのテロ資金調達対策（CFT）など、無数のユースケースをサポートするための高価な専用 AML ソフトウェアを購入する予算がないため、これは重要なことです。市場では、上記のグラフ解析を行うことができる専用ソリューション、ウェアハウスでの BI に対応する専用ソリューション、機械学習に対応する専用ソリューションがあります。AML レイクハウスのデザインは、この 3 つを統合したものです。AML データプラットフォームチームは、クラウドストレージの低コストで Delta Lake を活用しながら、オープンソーステクノロジーを容易に統合し、グラフテクノロジー、コンピュータビジョン、SQL アナリティクスエンジニアリングに基づくキュレーションレポートを作成できます。図 11 は、AML のレポーティングの具体的な例です。

付属の Notebook では、トランザクションオブジェクト、エンティティオブジェクトのほか、事前に学習させたモデルを使った見込み客の構造化、合成 ID 階層、アドレス分類などのサマリーが作成 されています。図 11 の Databricks SQL による可視化では、Databricks Photon SQL エンジンを使って、これらにサマリーを実行し、ビルトインの可視化により、数分でレポートダッシュボードを作成できました。ダッシュボード自体だけでなく、両方のテーブルに完全な ACL があり、ユーザーがエグゼクティブやデータチームと共有できるようになっています。このレポートを定期的に実行するスケジューラーも組み込まれています。このダッシュボードは、AML ソリューションに組み込まれた AI、BI、分析エンジニアリングの集大成です。

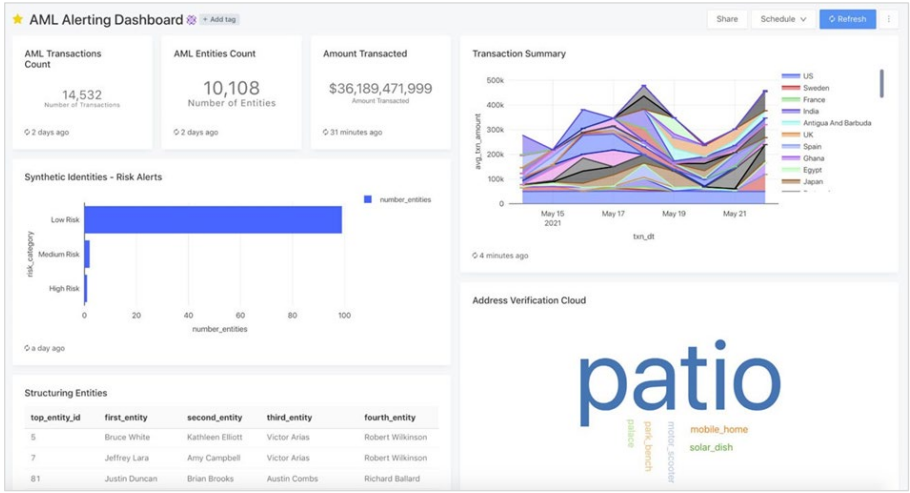


図 11

オープンバンキングの変革

オープンバンキングの台頭により、金融機関は API を通じて消費者、金融機関、サードパーティのサービスプロバイダ間でデータを共有し、優れた顧客体験の提供ができるようになりました。その一例が決済サービス指令（PSD2） で、欧州オープンバンキング規制の一環として EU 地域の金融サービスを変革しました。その結果、金融機関は複数の銀行やサービスプロバイダを通じて、顧客の口座や取引データなど、より多くのデータにアクセスできるようになりました。この傾向は、米国愛国者法第 314 条（b） に基づく FinCEN の最新のガイダンスにより、詐欺や金融犯罪の世界で拡大しています。対象となる金融機関は、他の金融機関や国内外の支店内で、マネーロンダリングの可能性が疑われる個人、企業、組織などに関する情報を共有できるようになりました。

情報共有の規定は透明性を高め、マネーロンダリングやテロ資金調達から米国の金融システムを保護するのに役立ちますが、情報交換は適切なデータとセキュリティ保護を備えたプロトコルを使用して行わなければなりません。情報共有の安全性という問題を解決するために、Databricks では最近、データ共有のためのオープンで安全なプロトコルである Delta Sharing を発表しました。Pandas や Spark といった使い慣れたオープンソースの API を使用することで、データの生産者と消費者はセキュアでオープンなプロトコルを使用してデータを共有し、あらゆるデータトランザクションを完全に監査して FinCEN 規制の遵守の維持が可能になりました。

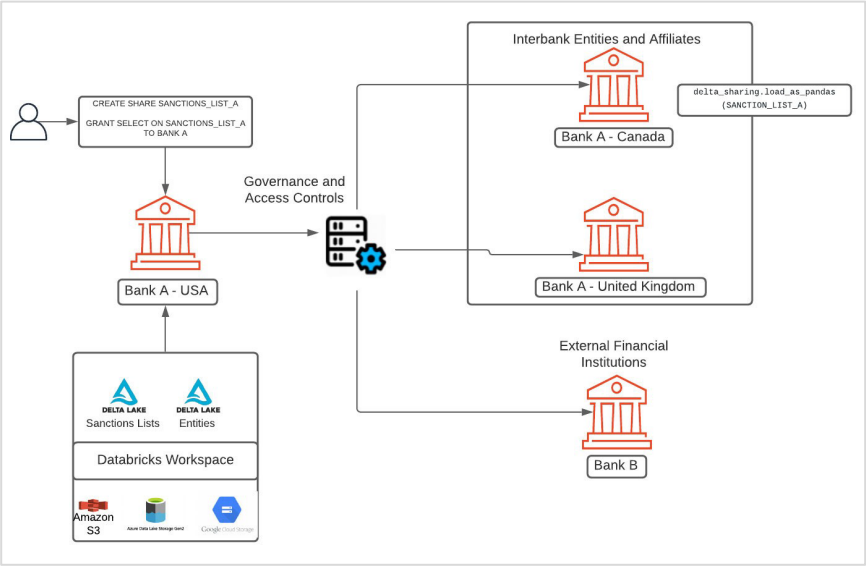


図 12

まとめ

レイクハウスアーキテクチャは、スケーラブルで汎用性の高いプラットフォームで、アナリストの AML 分析を可能にします。レイクハウスは、ファジーマッチ、画像分析、ダッシュボード内蔵の BI といった幅広いユースケースをサポートしており、これらの機能により、企業は独自の AML ソリューションと比較して総所有コストの削減が可能になります。Databricks の金融サービスチームは、金融サービス分野におけるさまざまなビジネス問題に取り組み、AML などのソリューションアクセラレータを通じて、データエンジニアリングやデータサイエンスの専門家が Databricks ジャーニーを始められるよう支援しています。

Databricks Notebook の無料お試し

- AML のためのグラフ理論概要
- AML のためのコンピュータビジョン概要
- AML 向けエンティティ解決概要

SECTION 2.6

ゲームにおける有害行為の検知のためのリアルタイム AI モデルの構築

投稿者：Dan Morris、 Duncan Davis

2021年6月16日

多人数参加型オンラインゲーム（MMO）や多人数参加型オンラインバトルアリーナゲーム（MOBA）などのオンラインゲームでは、プレイヤー同士がリアルタイムに交流し、協調したり競ったりしながら、共通の目標である勝利に向かって進んでいくことができます。このようなインタラクティブ性は、ゲームプレイのダイナミクスに不可欠ですが、同時に、オンラインゲーム界に蔓延する問題である有害行為の素地にもなっています。

有害な行動はさまざまな形で現れます。例えば、Behaviour Interactive 社のマトリックスには、マルチプレイヤーゲーム「Dead by Daylight」内で見られるインタラクションの種類が記載されており、グリーフ、ネットいじめ、セクハラのさまざまな程度が示されています。

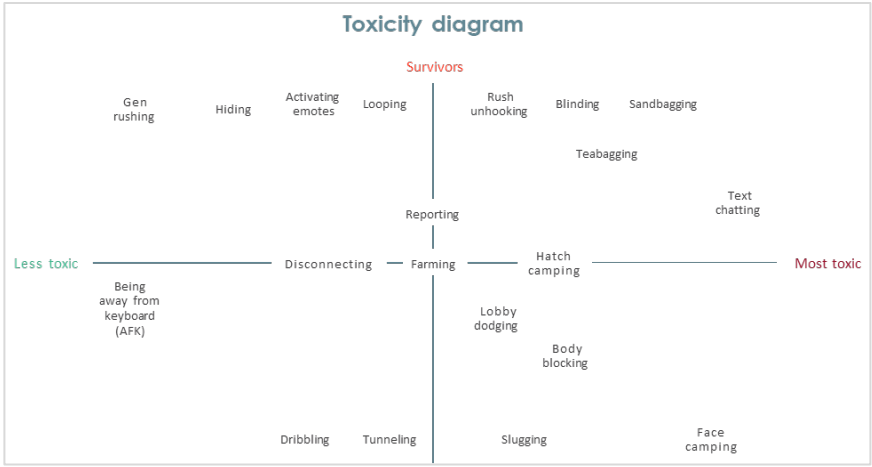


図1：プレイヤーが経験する有害な相互作用のマトリクス

有害行為がゲーマーやコミュニティに与える個人的な被害（深刻な問題）に加えて、多くのゲームスタジオの収益にも損害を与えています。例えば、ミシガン州立大学の研究の調査によると、80% のプレイヤーが有害性を経験し、そのうちの 20% がその相互作用が原因でゲームから離れたと報告しています。同様に、ティルブルグ大学の研究でも、ゲームの最初のセッションで破壊的または有害な出会いを経験した場合は、プレイヤーは 3 倍以上の確率でゲームを離れ、戻らなくなることが示されています。多くのスタジオにとって、プレイヤーの維持は最優先事項であり、特にゲームの配信が物理的なメディアリリースから長期的なサービスへと移行するなかで、有害性を抑制する必要があることは明らかです。

この離脱に関する問題をさらに深刻にするのが、開発の初期段階の発売前であっても毒性に関する問題に直面する企業です。例えば、Amazon のクルーシブルは、有毒なプレイヤーやインタラクションを監視または管理するシステムが導入されていないことも原因で、テキストチャットやボイスチャットなしでテストにリリースされました。これは、ゲーム空間の規模が、ほとんどのチームの管理能力をはるかに超えていることを示しています。このことから、スタジオは開発ライフサイクルの早い段階でゲームに分析を統合し、有害なインタラクションを継続的に管理できるように設計することが不可欠となります。

ゲームにおける有害性は、明らかにビデオゲーム文化の一部となっている多面的な問題であり、単一の方法で普遍的に対処できるものではありません。 しかし、ゲーム内チャットでの有害行為に対処することは、有害行為の頻度や、自然言語処理（NLP）による自動検出の可能性を考えると、大きな影響を与えることができます。

Databricks ソリューションアクセラレータ「ゲームにおける有害性の検知」

このソリューションアクセラレータでは、Jigsaw の有害コメントデータと、Dota 2 のゲームマッチングデータを使用し、NLP と既存レイクハウスを利用して、有害なコメントのリアルタイム検知に必要なステップを説明します。NLP については、このソリューションアクセラレータでは Apache Spark™ 上にネイティブに構築されたオープンソースのエンタープライズグレードのソリューション、ジョン・スノー・ラボの Spark NLP を使用しています。

このソリューションアクセラレータで行うステップは、次のとおりです。

- Delta Lake を使って Jigsaw とDota 2 のデータをテーブルに読み込む
- マルチラベル分類を用いた有害コメントの分類（Spark NLP）
- MLflow を用いた実験の追跡とモデルの登録
- バッチデータ、ストリーミングデータへの推論適用
- ゲームマッチデータにおける有害性の影響を検証する

本稼働環境のゲーム内チャットの有害性を検知

このソリューションアクセラレータを使用することで、容易に有害性検知機能をゲームに組み込むことができるようになります。図 2 のリファレンスアーキテクチャでは、ストリーム、ファイル、音声、運用データベースなど、さまざまなソースからチャットやゲームのデータを取得し、Databricks を活用してデータを取り込み、保存、キュレーションして機械学習（ML）パイプライン、ゲーム内機械学習、分析用の BI テーブル、さらにはコミュニティのモデレーションに使用するツールと直接対話する方法を示しています。

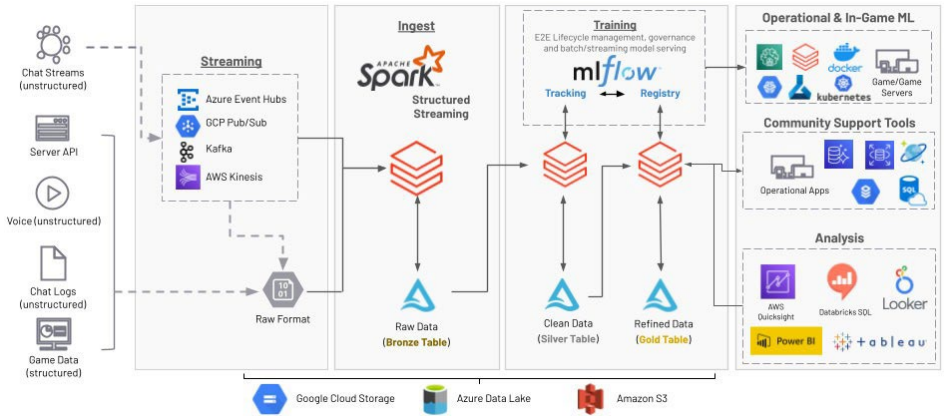


図 2：有害性検知のリファレンスアーキテクチャ

コミュニティの有害性を検出するためのリアルタイムでスケーラブルなアーキテクチャを持つことで、コミュニティのリレーションシップマネージャーのワークフローを簡素化し、数百万のインタラクションを管理可能なワークロードにフィルタリングできるようになります。同様に、深刻な有害事象をリアルタイムで警告したり、プレイヤーのミュートや CRM への迅速な通報といった対応を自動化することも、プレイヤーの定着に直接的な影響を与える可能性があります。同様に、異なるソースからの大規模なデータセットを処理できるプラットフォームがあれば、レポートやダッシュボードを通じてブランドの認知度を監視できます。

まずはここから

このソリューションアクセラレータの目的は、ゲーム内のチャットで有害な発言をリアルタイムに検知し、オンラインゲームにおける有害な交流の継続的な管理を支援することです。このソリューションアクセラレータを Databricks のワークスペースに直接インポートして、今すぐ始めましょう。

インポートすると、2 つのパイプラインを持つ Notebook が利用可能になり、本番環境に移行する準備ができます。

- Google Jigsaw の実世界英語データセットで学習したマルチラベル分類を用いた ML パイプライン：モデルは、テキスト中の毒性の形態を分類し、ラベル付けします。
- 有害性モデルを活用したリアルタイムストリーミング推論パイプライン：パイプラインソースは、一般的なデータソースからチャットデータを取り込むために簡単に変更できます。

この 2 つのパイプラインを利用すれば、最小限の労力で有害性の理解と分析を開始できます。また、このソリューションアクセラレータは、ゲームの仕組みやコミュニティへの関連データを用いてモデルを構築、カスタマイズ、改善するための基盤を提供します。

Databricks Notebook の無料お試しをご利用ください。

SECTION 2.7

Northwestern Mutual 導入事例：インサイトプラットフォームをスケーラブルでオープンなレイクハウスアーキテクチャに移行し、変革を推進

投稿者：Madhu Kotian （ノースウェスタン・ミューチュアル社 ）

2021年7月15日

デジタルトランスフォーメーション（DX）は、現代のほとんどのビッグデータ企業イニシアチブ、特にレガシーフットプリントが多い企業において、最前線で中心的な役割を担っています。DX の下支えとなるコンポーネントの一つが、データとそれに関連するデータストアです。 Northwestern Mutual（ノースウェスタン・ミューチュアル）は 160 年以上にわたり、家庭や企業が経済的な安定を得られるよう支援しています。310 億ドル以上の収益、460 万人以上の顧客、9,300 人以上の金融専門家を抱え、さまざまなソースでこれだけのデータ量を持つ企業はそう多くはないでしょう。

組織がさまざまなフォーマット、時間枠、さまざまな方向から集まる何百万ものデータポイントをかつてないほど大量に扱う今日において、データの取り込みは課題です。ノースウェスタン・ミューチュアルでは、データを意味あるものにし、分析に活用できる状態にすることを目指していました。このブログでは、データ取り込みプロセス、スケジューリングプロセス、そしてデータストアとの付き合い方を変革し、近代化するための斬新なアプローチをご紹介します。 私たちが学んだことのひとつは、効果的なアプローチは多面的であるということです。そこで、技術的なアレンジメントに加え、チームへの導入計画についてもご紹介します。

課題

私たちは変革に着手する前に、ビジネスパートナーと協力して技術的な制約を理解し、ビジネスケースの問題ステートメントの形成に役立てました。

特定したビジネス上の問題点は、顧客データやビジネスデータが社内外の異なるチームやデータソースから提供されており、統合されたデータがないことでした。リアルタイムデータの価値に気づいていましたが、タイムリーにビジネス上の意思決定を行えるような本番またはリアルタイムデータへのアクセスは限られていました。また、ビジネスチームが構築したデータストアは、データのサイロ化を招き、その結果、データの遅延問題、データ管理のコスト増加、不当なセキュリティ制約を引き起こしていることも分かりました。

さらに、現状に対する技術的な課題もありました。需要の増加と必要なデータの増加に伴い、インフラのスケーラビリティ不足、データ遅延、データサイロによる高い管理コスト、データのサイズと量の制限、データセキュリティの問題などがありました。このように課題が山積みとなり、私たちは、変革の成功に協力してくれる適切なパートナーを見つける必要があると考えました。

ソリューション分析

競争力を高め、お客様へのサービスを向上させ、社内プロセスを最適化するためには、データドリブンになる必要がありました。さまざまな選択肢を検討し、いくつかの POC を実施し、最終的な推奨事項を選定しました。今後の戦略として、次のものは必須でした。

- データ取り込み、データ管理、分析のニーズに応えるオールインワンソリューション
- 開発者やビジネスアナリストが SQL を使用して分析を行うことを効果的にサポートできる最新のデータプラットフォーム
- S3上で ACID トランザクションをサポートし、ロールベースのセキュリティを実現できるデータエンジン
- PII/PHI 情報を効果的に保護できるシステム
- データ処理と分析の需要に応じて自動的に拡張できるプラットフォーム

私たちのレガシーインフラは、MSBI Stack が基盤でした。インジェストには SSIS、データストアには SQL Server、Tabular モデルには Azure Analysis Service、ダッシュボードとレポート作成にはは PowerBI を使用していました。当初、このプラットフォームはビジネスのニーズを満たしていましたが、データ量とデータ処理需要の増加に伴う拡張性に課題があり、データ分析への期待が制限されていました。さらにデータニーズが高まるなか、ロード遅延によるデータレイテンシの問題や、特定のビジネスニーズに対応したデータストアが、データサイロやデータスプロールの原因となっていました。

複数のデータストアにデータが分散しているため、セキュリティも課題でした。日々のジョブでは、7 時間以上かかっている ETL ジョブが約 300 件あり、どのような変更、新規開発であっても、市場投入までの時間は約 4〜6 週間（複雑さにより異なる）かかっていました。

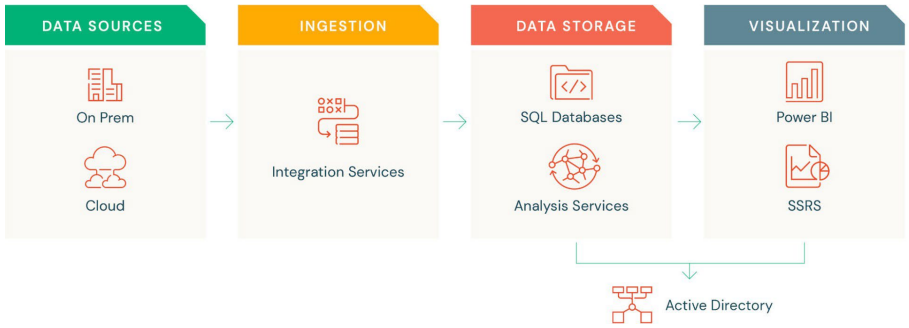


図1： レガシーアーキテクチャ

市場で複数のソリューションを評価した結果、オープンなレイクハウスアーキテクチャで単一の統合データ管理ソリューションを提供する Databricks の採用を進めることになりました。

Apache Spark™ 上で開発された Databricks により、Python を使用してデータ取り込みとメタデータ管理のためのカスタムフレームワークを構築できました。また、アドホック解析やノートブックによるデータ検索などの柔軟な実行も可能です。

Databricks の Delta Lake（データレイクの上に構築されたストレージ層）は、必要なセキュリティ制御の実装を含め、さまざまなデータベース管理機能（ACID トランザクション、メタデータガバナンス、タイムトラベルなど）を柔軟に実装できました。Databricks により、クラスタの管理やスケーリングの複雑さが排除され、エンジニアやビジネスユーザーから殺到する需要にも効果的に対応できるようになりました。

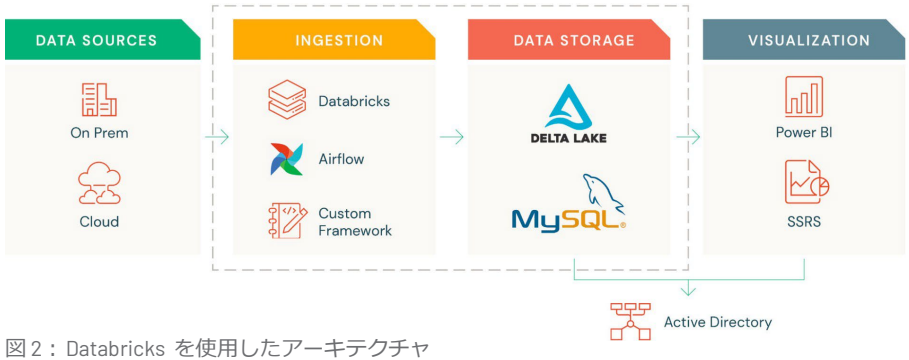


図 2：Databricks を使用したアーキテクチャ

移行アプローチとオンボーディングリソース

まず、少人数のエンジニアからスタートし、既存のスクラムチームからバーチャルチームにアサインしました。さまざまな POC を実行し、推奨されるソリューションの構築、ベストプラクティスの開発を行い、それぞれのチームに戻ってオンボーディングを支援することを目指しました。

既存のチームメンバーは、レガシーシステムの知識を持ち、現在の取り込みフローやビジネスルールを理解し、少なくとも 1つのプログラミング知識（データエンジニアリング＋ソフトウェアエンジニアリング）に精通していたため好都合でした。このチームは、Python のトレーニングを受け、Spark と Delta の詳細を理解し、Databricks チームと密接に連携してソリューションやアプローチの検証を行いました。このチームが将来の形成に取り組んでいる間、他の開発者はビジネスの優先事項の実現に取り組みました。

開発者のほとんどが MSBI Stack のエンジニアであったため、開発者、ビジネスユーザー、フィールドアドバイザーにスムーズなデータプラットフォームを提供することを目指しました。

- データのロードと変換のニーズを全てカバーするインジェストフレームワークを構築しました。セキュリティ制御が組み込まれており、ソースシステムの全てのメタデータと機密を保持できました。インジェストプロセスでは、ソース、ターゲット、必要な変換を含む JSON ファイルを採用することで、単純な変形と複雑な変形の両方を可能にしました。
- スケジューリングには最終的に Airflow を使用しましたが、DAG が複雑なため、Airflow の上に独自のフレームワークを構築し、ジョブ情報とその相互依存性を含む YAML ファイルを採用しました。
- Delta を使用したスキーマレベルの変更管理には、独自のカスタムフレームワークを構築し、開発者がデータストアにブレークグラスでアクセスすることなく、さまざまなデータベース型操作（DDL）を自動化できました。これは、データストアにさまざまな監査コントロールを導入するのにも役立ちました。

また、並行してセキュリティチームと連携し、データセキュリティの基準（転送時、保存時、PII 情報保護のための列レベルの暗号化）を全て理解し、基準を満たしているかどうかを確認しました。

フレームワークがセットアップされると、コホートチームはエンドツーエンドのフロー（ソースからターゲットへの全ての変換）を展開し、PowerBI 上で Delta Lake を指す新たなレポート、ダッシュボードを生成しました。エンドツーエンドプロセスの性能テスト、データの検証、そして現場のユーザーからのフィードバックを得ることが目的です。性能、検証テストのフィードバックや結果をもとに、段階的に改良を加えていきました。

同時に、開発者のトレーニングガイドとハウツーを作成し、開発者の育成に努めました。その後すぐに、コーホートメンバーをそれぞれのチームに移し、数名を残してプラットフォーム基盤のサポート（DevOps）を継続しました。各スクラムチームは、それぞれのケイパビリティや機能セットを管理し、ビジネスに反映する責任がありました。チームメンバーがそれぞれのチームに戻ると、移行作業のバックログを含めるようにチームの速度を調整するタスクに着手しました。チームリーダーには、さまざまなスプリントやプログラムの増分の移行目標を達成するために、具体的なガイダンスと適切な目標が与えられていました。 コーホートグループにいたメンバーは、常駐するエキスパートとなり、チームの新しいプラットフォームへのオンボードを支援しました。データブリックスはその場限りの質問やサポートにも対応してくれました。

新しいプラットフォームを段階的に構築していく過程で、検証、確認のために旧プラットフォームを残しました。

成果の始まり

全体的な変革には約 1 年半かかりました。全てのフレームワークの構築、ビジネスの優先順位の管理、セキュリティ要件への対応、チームの再編成、プラットフォームの移行をしなければならなかったことを考えると、かなりの偉業です。全体のロード時間は、7 時間からわずか 2 時間に短縮され、市場投入までの期間は、4〜6 週間から、およそ 1〜2 週間に大幅に短縮されています。これは大きな改善であり、これからのビジネスにさまざまな形で貢献していくことでしょう。

私たちの変革はまだ終わっていません。プラットフォームの充実を図りながら、次のミッションはレイクハウスパターンの拡張です。E2 へのプラットフォームの移行と Databricks SQL の導入に取り組んでいます。アドホック分析を可能にするセルフサービスプラットフォームをビジネスユーザーへ提供し、ビジネスユーザーが独自のデータを持ち込み、私たちの統合データを使って分析を実行できる機能を持つための戦略を進めています。オープンで統合されたスケーラブルなプラットフォームを使用することで、大きな利益を得ることができることを学びました。ニーズとケイパビリティが高まったとしても、私たちにはデータブリックスという強力なパートナーがいます。

ノースウェスタン・ミューチュアルのレイクハウスへのジャーニーについて詳しくは、こちらでご覧いただけます。

Madhu Kotians 氏について

Madhu Kotian 氏は、ノースウェスタン・ミューチュアル社のエンジニアリング（投資商品データ、CRM、アプリ、レポーティング）担当VP です。IT 分野で 25 年以上の経験を持ち、データエンジニアリング、人材管理、プログラムマネジメント、アーキテクチャ、設計、アジャイルプラクティスを用いた開発・保守の経験と専門知識を持っています。また、データウェアハウスの方法論とデータ統合と分析の実装の専門家でもあります。

SECTION 2.8

3つのクラウドと50以上のリージョンにまたがるDatabricksレイクハウスの構築

投稿者：Jason Pohl、Suraj Acharaya

2021年7月14日

データブリックスは、内部ロギングフラを長期にわたって進化させています。その過程で、複数のクラウドとリージョンをまたいで高可用性のログパイプラインを維持する方法について、いくつかの教訓を得ました。本ブログでは、レイクハウスプラットフォームを使用したリアルタイムのメトリクスの収集・管理と、複数のクラウドを活用した、パブリッククラウドの障害からの回復方法についてのヒントをご紹介します。

創立当初は、1つのパブリッククラウドにしか対応していませんでした。現在では、世界50以上の地域で3大パブリッククラウド（AWS、Azure、GCP）をサポートするサービスに成長しました。データブリックスは日々、お客様に代わって数百万の仮想マシンを稼働させています。10人弱のエンジニアからなるデータプラットフォームチームは、1日あたり0.5ペタバイトのデータを処理するロギングテレメトリインフラの構築と保守を担っています。オーケストレーション、モニタリング、および使用状況は、インフラで処理されるサービスログを介して取得され、タイムリーで正確な測定基準が提供されます。最終的に、このデータは私たち独自のペタバイトサイズのDelta Lakeに保存されます。データプラットフォームチームは、Databricksを使用してクラウド間処理を行い、必要に応じてデータを連携させ、リージョンにおけるクラウド停止からの復旧を支援し、稼働中のインフラへの影響を最小限に抑えています。

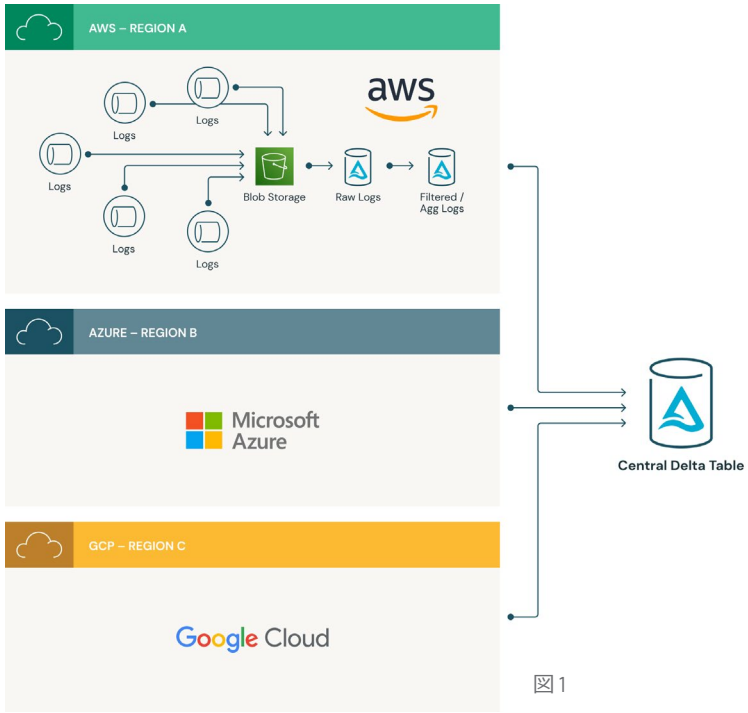


図1

パイプラインアーキテクチャ

各クラウドリージョンには、ログデータをキャプチャ、収集、リージョンのDelta Lakeに保存するための独自のインフラとデータパイプラインが含まれています。製品の遠隔測定データは、全てのクラウドリージョンにわたって複製された同じプロセスによって、製品全体とパイプライン内でキャプチャされます。ログデーモンは、テレメトリデータをキャプチャし、これらのログをリージョンのクラウドストレージバケット（S3、WASBS、GCS）に書き込みます。そこから、スケジュールされたパイプラインは、オートローダ（AWS | Azure | GCP）を使用してログファイルを取り込み、データをリージョンのDeltaテーブルに書き込みます。別のパイプラインは、リージョンのDeltaテーブルからデータを読み取り、フィルタリングして、単一のクラウドリージョンの集中型Deltaテーブルに書き込みます。

Delta Lake 以前

Delta Lake が導入される前は、ソースデータを集中型レイク内の独自のテーブルに書き込み、全てのテーブルをまたいでユニオンとなるビューを作成していました。このビューは実行時に計算する必要があり、リージョンを追加するにつれて非効率的になりました。

```
CREATE OR REPLACE VIEW all_logs AS
SELECT * FROM (
   SELECT * FROM region_1.log_table
   UNION ALL
   SELECT * FROM region_2.log_table
   UNION ALL
   SELECT * FROM region_3.log_table
   ...
);
```

Delta Lake 導入後

現在では、50 以上の異なるリージョンから同時に書き込みステートメントを受け入れる単一の Delta テーブルを持っているだけです。データに対するクエリを同時に処理します。これにより、中央テーブルのクエリは次のように簡単になります。

```
SELECT * FROM central.all_logs;
```

トランザクションは Delta Lake で処理されます。私たちは、Delta Lake にある個々のリージョンテーブルを非推奨とし、UNION ALL ビューを廃止しました。次のコードは、リージョンの Delta Lake から中央の Delta Lake への出力が承認されたデータをロードするために実行される構文を簡略化して表したものです。

```
spark.readStream.format("delta")
   .load(regional_source_path)
   .where(“egress_approved = true”)
   .writeStream
   .format("delta")
   .outputMode("append")
   .option("checkpointLocation", checkpoint_path)
   .start(central_target_path)
```

ディザスタリカバリ

クラウド間サービスを運用する利点の1つは、特定のディザスターリカバリーシナリオに備えることができる点です。稀なケースですが、特定のクラウドリージョンのコンピュートサービスが停止することは、決して珍しいことではありません。その場合、クラウドストレージにはアクセスできますが、新たな仮想マシンを立ち上げることはできません。データパイプラインのコードは、ソースパスとデスティネーションパスの設定を受け入れるように設計されているので、データが保存されている場所とは異なるリージョンでデータパイプラインを迅速にデプロイして実行できます。クラスタがどのクラウドに作られるかは、データの読み書きが行われるクラウドには関係ありません。

クラウドプロバイダ間で継続的にデータを複製することで、ストレージサービスの障害から保護するデータセットがいくつかあります。これは、ブログで紹介している Delta ディープクローンの機能を利用して簡単に実現できます。クローンコマンドをテーブルに対して実行するたびに、前回実行した時からの増分変更のみでクローンが更新されます。これは、リージョンやクラウド間でさえもデータを複製する効率的な方法です。

ライブデータパイプラインの中断を最小限に抑える

データパイプラインは、マネージドサービスの生命線であり、眠らないグローバルビジネスの一部でもあります。メンテナンス、アップグレード、データの埋め戻しなどのために、パイプラインを長期間停止する余裕はありません。ついこの間、パイプラインをフォークして、通常はメインテーブルに書き込まれるデータのサブセットをフィルタリングし、別のパブリッククラウドに書き込む必要がありましたが、通常通りのビジネスを中断させることなく行うことができました。

以下の手順を踏むことで、アーキテクチャの変更を本番システムに導入する際にも、混乱が生じないようにすることができました。

> まず、メインテーブルのディープクローンを、もう一方のクラウド上の新しいロケーションに実行しました。これにより、データとトランザクションログの両方が、一貫性を保つようにコピーされます。

> 次に、新しい設定をパイプラインにリリースしました。データの大部分は引き続き中央のメインテーブルに書き込まれ、データのサブセットは別のクラウドにある新しいクローンテーブルに書き込まれるようにしました。この変更は、新しい設定をデプロイするだけで、容易に行うことができ、テーブルは、受け取るべき新しい変更だけの更新を受け取ります。
>
> 次に、同じディープクローンコマンドを再度実行しました。Delta Lake は、元のメインテーブルから新しいクローンされたテーブルへの増分変更のみを取り込み、コピーします。これは基本的に、ステップ1と2の間のデータへの全ての変更で新しいテーブルを埋め戻すものです。
>
> 最後に、メインテーブルからデータのサブセットを削除し、クローンテーブルからデータの大部分を削除することができます。これで、両方のテーブルが、トランザクションの履歴を全て含む、本来のデータを表すようになり、パイプラインの鮮度を損なうことなく、ライブで実行されました。

まとめ

Databricks は、クラスタマネージャによるインフラストラクチャのスピンアップ、オートローダによるデータの取り込み、Delta Lake によるクラウドストレージへのトランザクション書き込みなど、個々のクラウドサービスの詳細を抽象化しています。このため、データ連携やディザスタリカバリのために、パブリッククラウド上のコンピュートとストレージを単一のコードベースでブリッジできる利点があります。このクラウド間機能により、私たちとお客様に最適な場所にコンピュートとストレージを移動させる柔軟性が生まれます。

SECTION

03 導入事例

- アトラシアン（Atlassian）
- ABM アムロ銀行（ABN AMRO）
- J.B. ハント（J.B. Hunt）

SECTION 3.1

アトラシアン（Atlassian）

発祥の SaaS 企業アトラシアン（Atlassian）は、チームコラボレーション、開発、課題管理ソフトウェアのリーディングプロバイダとして、 Jira、Confluence、Bitbucket、Trello などの製品を通じて、フォーチュン 100 社のうちの 85 社を含む 15 万社以上のグローバルな顧客におけるコラボレーションを支援しています。

ユースケース

アトラシアンでは、Databricks レイクハウスプラットフォームを活用し、企業全体におけるデータの民主化、運用コストの削減に成功しています。アトラシアンは現在、顧客体験の改善を中心としたユースケースを多数有しています。

カスタマーサポート、サービス経験

Jira や Confluence などのサーバーベースの製品を使用している大半の顧客をクラウドに移行し、顧客サポート体験を向上させるための深い知見の活用に着手しました。

マーケティングパーソナライゼーション

分析から得た知見を利用して、パーソナライズされたマーケティングメールを配信し、新機能や新製品のエンゲージメントを促進させています。

不正使用の防止と不正行為の検知

異常検知と予測分析により、ライセンスの不正使用や不正行為を予測できます。

「常に進化する目標を達成するには、チーム間のコラボレーションが欠かせません。シンプルなレイクハウスアーキテクチャは、膨大な量のユーザーデータの取り込み、顧客ニーズのより的確な予測および顧客体験改善に向けた分析の基盤となっています。使いやすい単一のクラウド分析プラットフォームが、実践的な知見に基づく新たなコラボレーションツールの迅速な改善と構築を可能にしています。」

アトラシアン
データプラットフォーム部門シニアマネージャー
Rohan Dhupelia 氏

ソリューションと効果

アトラシアンは、Databricks レイクハウスプラットフォームを利用して、社内外を問わず大規模なデータの民主化を実現しています。データウェアハウスのパラダイムから Databricks の標準化へと移行し、データドリブンな組織となりました。人事、マーケティング、財務、研究開発など、組織の半分以上にあたる 3,000 名を超えるの社内ユーザーが、Databricks SQL などのオープンテクノロジーを通じて、毎月このプラットフォームから分析結果にアクセスしています。また、アトラシアンではこのプラットフォームを利用して、顧客にパーソナライズされたサポートとサービス体験を提供しています。

- Delta Lake によって単一のレイクハウスが基盤の Delta Lake により、人事、マーケティング、財務、研究開発など 3,000 名以上のユーザーがペタバイト規模のデータにアクセスできるようになった。
- Databricks SQL を活用した BI ワークロードにより、より多くのユーザーがダッシュボードレポートを作成できるようになった。
- MLflow によって MLOps が効率化し、納期が短縮した。
- データ統合プラットフォームがガバナンスを容易にし、セルフマネージドクラスタが自律性を促進させた。

クラウドスケールのアーキテクチャ、チーム間のコラボレーションの強化による生産性向上、分析や機械学習のための顧客データへのアクセスなど、アトラシアンでは、Databricks の導入による多大な効果が期待されています。現時点で既に次のような効果が実現しています。

- 最小限の労力とローコード変更で 5 万 件以上の Spark ジョブを EMR から Databricks に移行し、IT 運用コスト（特にコンピュートコスト）を 60% 削減した。
- 開発サイクルの短縮により、提供が 30% 迅速化した。
- 組織全体でセルフサービスが促進されたことで、データチームへの依存度が 70% 低減した。

詳しく見る

SECTION 3.2

ABN アムロ銀行（ABN AMRO）

オランダ金融大手の ABN アムロ銀行では、業務のモダナイズを目指すなか、従来型のインフラやデータウェアハウスによるデータソースへのアクセスの複雑化、非効率的なデータプロセスやワークフローが課題となっていました。ABN アムロ銀行は、これらの課題を解決すべく、Azure Databricks を導入。その結果、データと AI の民主化が加速し、500 名を超えるデータエンジニア、サイエンティスト、アナリストの連携が可能になり、全社的な業務の改善、新たな市場への参入が実現しています。

ユースケース

ABN アムロ銀行では、Databricks レイクハウスプラットフォームを活用した金融サービスにおけるグローバル規模のデジタル変革により、業務全体の自動化と知見抽出を促進しています。

金融のパーソナライズ

リアルタイムデータと顧客インサイトを活用し、顧客のニーズにあわせた商品とサービスを提供しています。例えば、自動化されたマーケティングキャンペーンでは、機械学習を利用してターゲットを絞ったメッセージングを行い、エンゲージメントとコンバージョンの向上に役立てています。

リスクマネジメント

データドリブンな意思決定により、企業と顧客の両方のリスクを軽減することに重点を置いています。例えば、ビジネスへの影響を回避するために、社内の意思決定者がリーダーがリスクを理解できるレポートやダッシュボードを作成しています。

不正行為の検知

悪意のある行為の防止を目的に、予測分析の活用により、顧客に影響が及ぶ前に不正行為を特定しています。検知対象となる行為には、マネーロンダリングやクレジットカードの不正な申請などが含まれます。

「Databricks の導入により、事業運営の方法が一変しています。データのスペシャリスト全員が、制御されたスケーラブルな環境で高度なデータ機能を利用できるようになり、企業としてデータと AI による変革を成功させる可能性が高まりました。」

ABN アムロ銀行
アナリティクス・エンジニアリング部門責任者
Stefan Groot 氏

ソリューションと効果

ABN アムロ銀行では、Azure Databricks を導入したことで、データと AI の民主化が加速し、500 名を超えるデータエンジニア、サイエンティスト、アナリストの連携が可能になり、全社的な事業運営の効率化、市場開拓の能力の向上が実現しています。

- Delta Lake によって高速で信頼性の高いデータパイプラインの構築が可能になり、ダウンストリーム分析に対して正確かつ完全なデータを供給できるようになった。
- Power BI との統合によって SQL による分析が容易になり、500名を超えるビジネスユーザーが、レポートやダッシュボードを通じて知見を抽出できるようになった。
- MLflow によって顧客体験を向上させる新しいモデルの展開が加速し、新しいユースケースを 2 か月以内に展開できるようになった。

10 倍迅速化
ユースケースを 2 か月で展開、市場投入を 10 倍迅速化

100＋
100 件以上のユースケースを今後 1 年間で展開

500＋
500 名以上の ビジネスおよび IT ユーザーが活用・連携

詳しく見る

SECTION 3.3

J.B. ハント（J.B. Hunt）

「Databricks の導入により、AI を活用した効果的な運送業者体験の提供が可能になり、そのことが、革新的でデジタルな輸送マーケットプレイスの中核となっています。」

J.B. ハント
エンジニアリング&テクノロジー部門ディレクター
Joe Spinelle 氏

詳しく見る

米トラック輸送大手の J.B.ハントは、北米で最も効率的なデジタル輸送ネットワークを構築するというミッションのもと、貨物物流を効率化し、運送業者体験の向上を目指していました。しかし、AI ケイパビリティの欠如、ビッグデータをセキュアに扱えないことなど、従来型のアーキテクチャにおける制限が大きな障壁となっていました。 Databricks レイクハウスプラットフォームと Immuta の導入により、サプライチェーンの効率化やドライバーの生産性向上といった、広範な業務ソリューションの提供が可能になり、IT インフラコストの大幅な削減と収益の向上に成功しています。

ユースケース

Carrier 360 プラットフォームに Databricks を活用し、業界屈指の輸送分析によるコスト削減とドライバーの生産性・安全性の向上を実現しています。Carrier 360 を用いた輸送・物流の最適化、顧客 360 度分析、パーソナライズなどのユースケースを展開しています。

ソリューションと効果

J.B.ハントでは、Databricks レイクハウスプラットフォームを活用して北米で最も安全かつ効率的な輸送マーケットプレイスを構築し、物流の効率化、運送業者体験の最適化、コスト削減を可能にしています。

- Delta Lake によるデータの統合と民主化：Carrier 360* でのリアルタイムなルート最適化、ドライバーレコメンデーション
- Notebook による生産性向上：データチームの生産性を高め、ユースケースの構築を促進
- MLflow による効率化：ドライバー体験を向上させる新たなモデルの展開を加速

270万ドル
IT インフラコストを削減

5%
ロジスティクスの改善による増収

99.8% 迅速化
レコメンデーションを 99.8% 迅速化、運送業者体験を改善

データブリックスについて

データブリックスは、米国サンフランシスコに本社を置き、世界中に拠点を持つデータと AI の企業です。Apache Spark、Delta Lake、MLflow のオリジナル開発メンバーによる創業以来、データの活用によって難題解決に挑む組織の支援に取り組んできました。Databricks レイクハウスプラットフォームは、コムキャスト（Comcast）、コンデナスト（Condé Nast）、H&M をはじめ、フォーチュン 500 企業の 40% を含むさまざまな業界の 5,000 社以上におけるデータ、分析、AI の取り組みに活用されています。

Twitter、LinkedIn、Facebook での情報発信も行っております。

カスタムデモのご予約

Databricks 無料お試し

databricks