BigQuery 継続的クエリと Vertex AI を活用したリアルタイム レコメンドシステムの構築

Google Cloud Next Tokyo

河西 隼太郎

合同会社DMM.com
開発統括本部
データ基盤開発部
データアプリケーショングループ
ML基盤チーム チームリーダー

上田 亮

合同会社DMM.com
開発統括本部
データ基盤開発部
データアプリケーショングループ
ML基盤チーム

目次

01. DMM について

なんでもやってる

AIから地方創生まで ——

16の領域で60以上の事業を運営

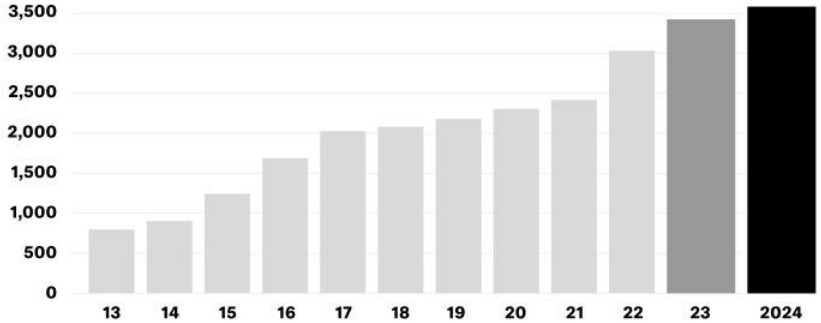
DMM

数字で見るDMMグループ

GROUP IN FIGURES

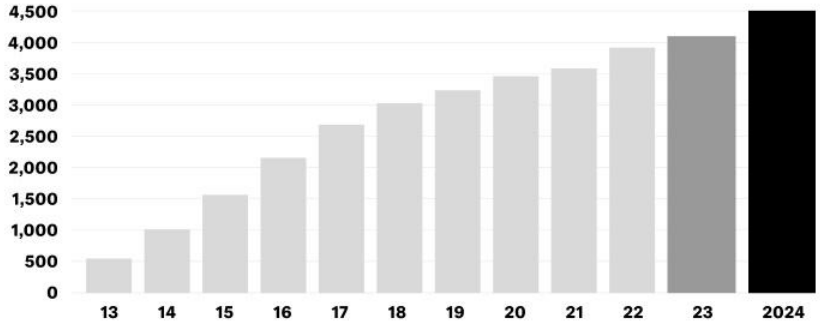
売上

3,637億円 ※1



会員数

4,507万人 ※2



事業数

60事業以上

グループ会社

24社

創業

27年目

グループ従業員

5,081名

※1 ： DMM.com、DMM.com 証券、DMM.com BASE、他連結（2月期）

※2 ： DMM.com サービスの会員数（2月期）

主な事業

DMM TV

月額550円のDMMプレミアム会員に登録することでアニメ約5,900作品を中心に19万本以上※1の映像作品や、漫画、2.5次元、声優にフォーカスしたオリジナル番組など多彩なエンタメコンテンツを楽しめるサブスクリプション動画配信サービス。

DMMブックス

話題のコミック、雑誌、小説、写真集等の電子書籍など126万冊以上を、スマホやパソコンで読めるプラットフォーム。

DMM pictures

日本が世界に誇るコンテンツ「アニメ」の企画開発、ライセンスビジネスや製作委員会への参画。

DMM STAGE

2.5次元作品を中心とした舞台を制作。 DMM picturesやDMM GAMESなど自社エンターテインメント領域のコンテンツを、グループシナジーを用いて舞台化。舞台以外にもLIVE、映像作品などを展開。

DMM GAMES

ユーザーに快適にゲームを楽しんでもらえるプラットフォームづくりや、ユニークなゲームの開発・パブリッシングを行っています。

DMMオンクレ

スマートフォンやパソコンを使って実物のクレーンゲーム機を遠隔操作し、24時間どこからでもクレーンゲームを楽しめるサービスです。

DMMスクラッチ

限定エンタメグッズやお得な雑貨・家電が当たるハズレなしのオンラインくじ。

DMMくじ

店頭で1枚から購入することができ、アニメやゲームをはじめ幅広いジャンルのグッズがその場で当たる、ハズレなしのくじ。

DMM Factory

キャラクターグッズ・フィギュア商品の企画開発/販売を行っています。

DMM FILMS

実写映画・ドラマ作品の企画・製作・出資を起点に、グループシナジーを活かして作品を多角的に楽しめる体験を提供する映像製作事業です。

DMM通販

DVD・Blu-ray、CD、本・コミック、ホビー、玩具、家電、日用品など、豊富な商品を展開。DMMによる仕入れ販売に加えて、法人や個人が出品できるサービスも提供し、コレクター商品など、希少性の高い商品も。

DMM宅配レンタル

DVD・Blu-ray／CD／コミックの宅配レンタルを送料無料で展開。

※1 ： 2024年4月時点

02. 課題と背景

レコメンドとは

レコメンドとは？

『あなたへのおすすめ作品』など。
ユーザーの閲覧・購買履歴や作品属性などに基づいて、そのユーザーに合うと考えられる作品などを先頭に並び替えてレコメンドする。

DMM での活用例

サービス：DMM TV に代表される SVOD サービス、DMM ブックスに代表される PPV サービス をはじめとした複数サービスに導入
設置面　：トップページ、作品の末端（詳細）ページ、メールマガジン

チーム体制

それぞれのチームが独立して業務を遂行

- インフラ構築
- API 開発
- 監視

MLOps エンジニア

- ML パイプライン開発
- ML モデル開発
- 評価分析

ML エンジニア

プラットフォーム提供

従来のレコメンド システム

データ更新は 1 日 1 回

- バッチ処理でレコメンド結果を更新
- ユーザーの行動が反映されるのは **1 日後**
- レコメンドは**過去**の興味関心に基づく

従来のレコメンド システム（改善点）

1

サイト滞在中にレコメンド反映

- 検索や閲覧などの行動を即座に反映する

2

現在の興味関心とのズレを最小限に

- 前回の訪問から時間が経過したユーザーに対してのズレを少なくする

3

新規ユーザーへのレコメンド

- 新規ユーザーへもレコメンドを提示できるようにする

従来のレコメンド システム（改善点）

1

サイト内行動をリアルタイムにレコメンド反映

- 検索や閲覧など直近の行動を即座に反映する

2

現在の興味関心とのズレを最小限に

前回の訪問から時間が経過したユーザーに対してのズレを少なくする

3

新規ユーザーへのレコメンド

- 新規ユーザーへもレコメンドを提示できるようにする

リアルタイム レコメンドが有効に働きそうな u2i (user to item) で検討を進める

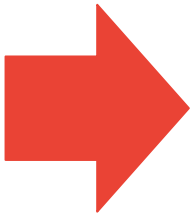
リアルタイム レコメンド システム検討

1. サイト滞在中のレコメンド反映
2. 現在の興味関心とのズレ
3. 新規ユーザーへのレコメンド
4. ML エンジニアと MLOps エンジニアの責任分界点
5. 素早いプロダクト ローンチ

- ユーザー課題
- 組織課題

リアルタイム レコメンド システム検討

1. サイト滞在中のレコメンド反映
2. 現在の興味関心とのズレ
3. 新規ユーザーへのレコメンド

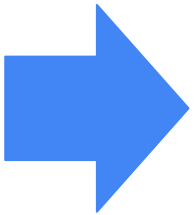


- ✓ サイト内ユーザーの行動をリアルタイムに連携
- ✓ Google Analytics 4 から BigQuery への連携

リアルタイム レコメンド システム検討

4 ML エンジニアと MLOps エンジニアの責任分界点

5 素早いプロダクトローンチ

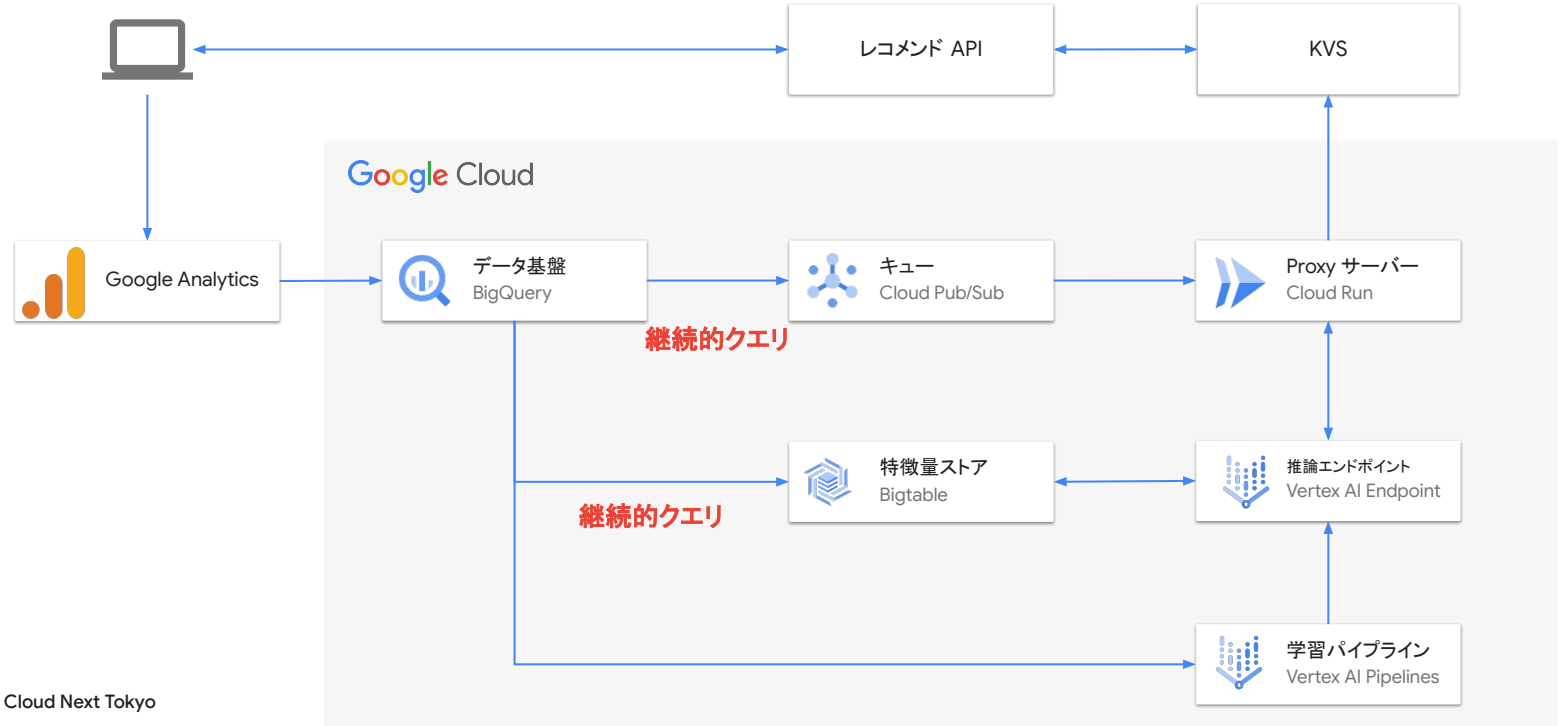


✓ ML エンジニアと MLOps エンジニアが扱うコンポーネントを明確にする

✓ 安全に素早く開発を行う

03. アーキテクチャ

アーキテクチャの全体像



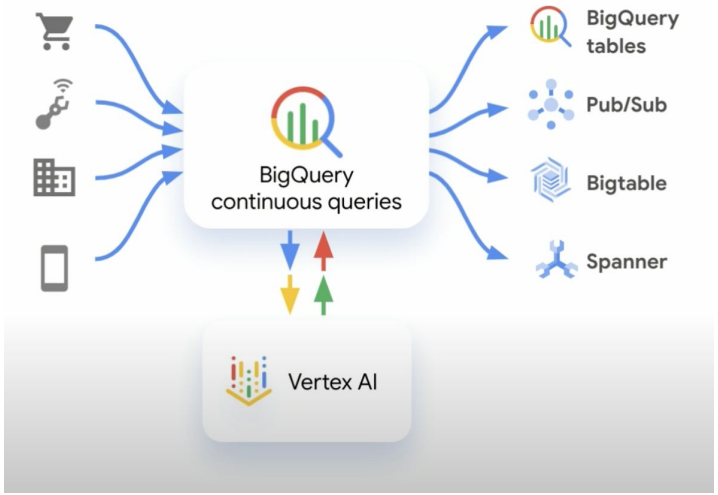
BigQuery 継続的クエリとは

継続的な SQL 実行

- SQL を BigQuery 上で継続的に実行し、リアルタイムなデータ変換を実現

画期的なポイント: BigQuery がストリーム処理エンジンに

- BigQuery へのストリーミング データの取り込みは従来から可能
- 継続的クエリの革新性 **リアルタイムで BigQuery に入ってくるデータを、ほぼ同時に変換・抽出し、外部サービスにエクスポートできる**



なぜ継続的クエリを採用したか

継続的クエリはプロジェクト当初の　2025 年 1 月時点では Preview 機能

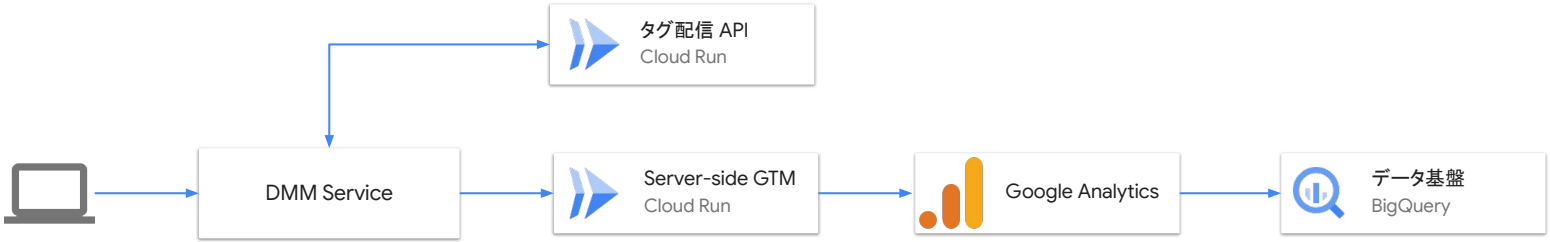
1. GA4 連携によるリアルタイムデータ基盤の存在

2. SQL でストリーミング処理を実装可能

なぜ継続的クエリを採用したか

1 GA4 連携によるリアルタイム データ基盤の存在

- GA4 -> BigQuery のストリーミング エクスポート機能
  - イベントが数秒 ~ 1 分程度で BigQuery に連携
- 内製の共通タグ配信 API + Server-side GTM
  - 全サービスで同一スキーマ ＋ DMM 独自パラメータを付与
  - -> 各サービスへの横展開が容易

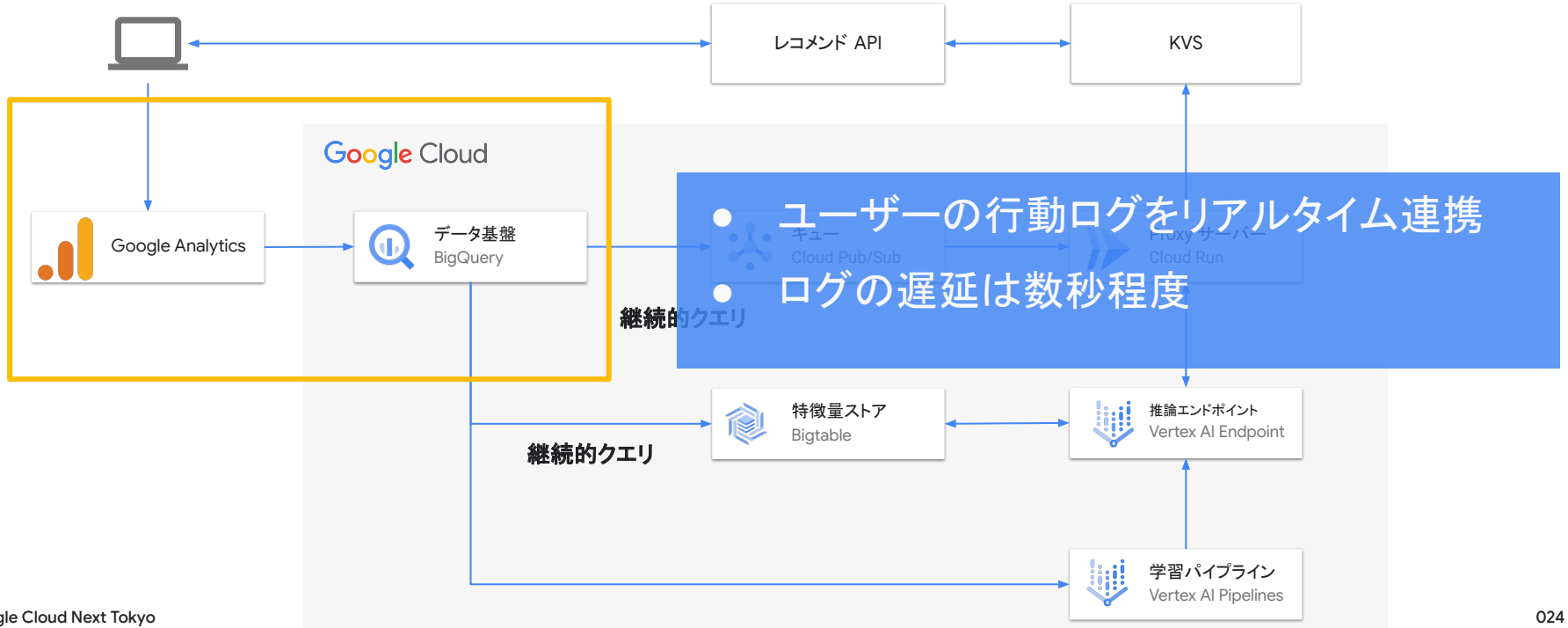


なぜ継続的クエリを採用したか

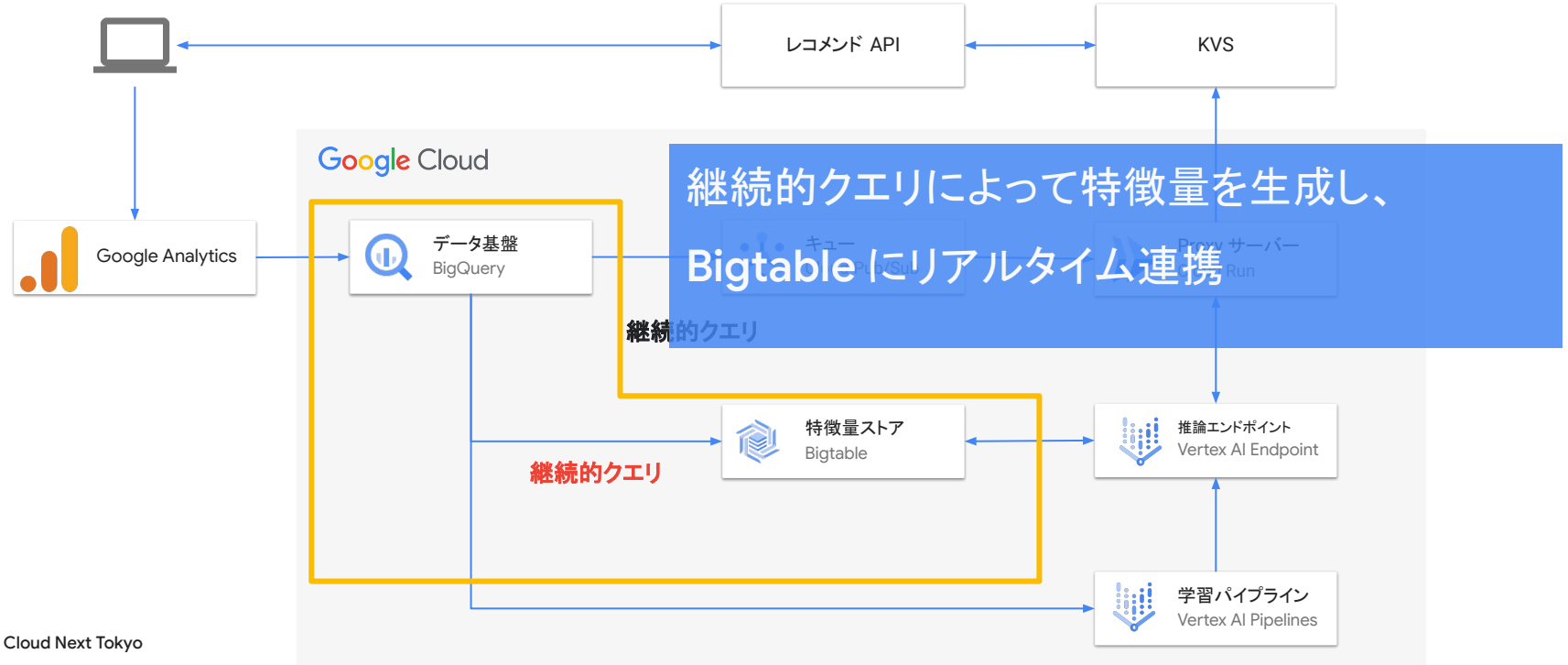
2 SQL でストリーミング処理を実装可能

- ML エンジニアが使い慣れた SQL だけで完結
  - 複雑になりがちなストリーミング処理フレームワークの学習コストが不要
- 開発のアジリティ向上
  - ML エンジニアが一人で高速にストリーミング パイプラインを実装・改善できる
  - MLOps エンジニアとのコミュニケーション コストがかからない

アーキテクチャ詳細

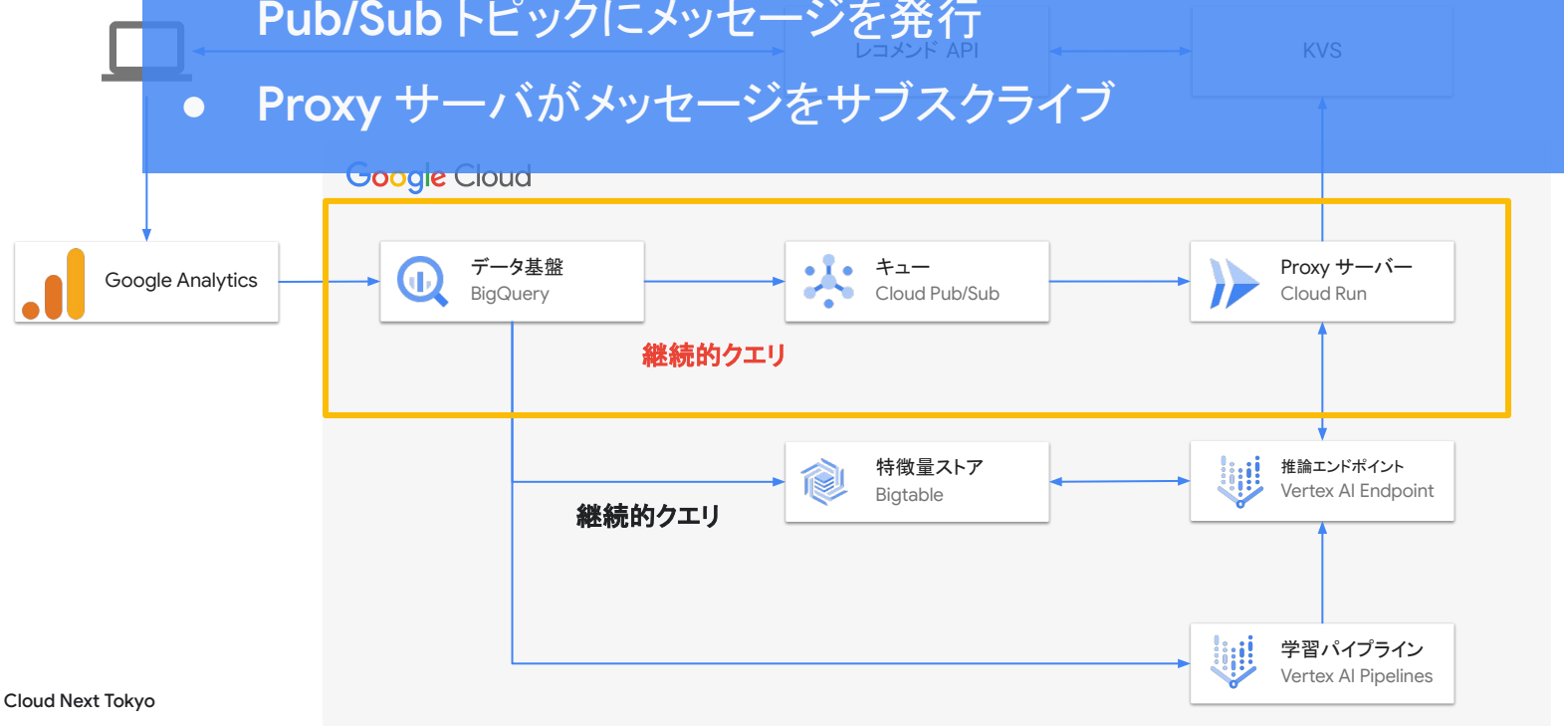

レコメンド API
KVS
Google Cloud
Google Analytics
データ基盤
BigQuery
継続的クエリ
キュー
Cloud Pub/Sub
Proxy サーバー
Cloud Run
ユーザーの行動ログをリアルタイム連携
ログの遅延は数秒程度
継続的クエリ
特徴量ストア
Bigtable
推論エンドポイント
Vertex AI Endpoint
学習パイプライン
Vertex AI Pipelines

アーキテクチャ詳細


レコメンド API
KVS
Google Cloud
継続的クエリによって特徴量を生成し、
Bigtable にリアルタイム連携
Google Analytics
データ基盤
BigQuery
継続的クエリ
特徴量ストア
Bigtable
推論エンドポイント
Vertex AI Endpoint
継続的クエリ
学習パイプライン
Vertex AI Pipelines

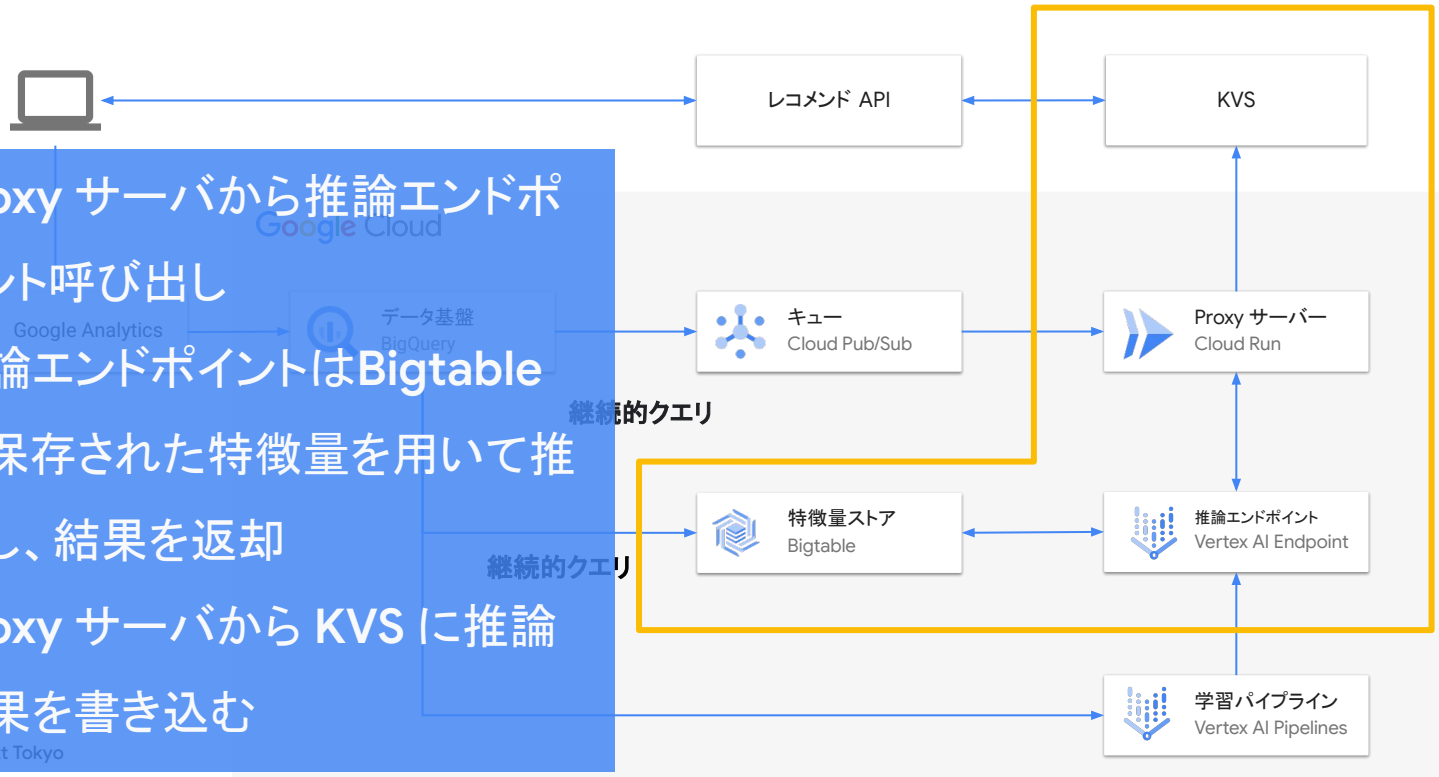
アーキテクチャ詳細

- 継続的クエリによって、ユーザーの行動ログの発生をトリガーに Pub/Sub トピックにメッセージを発行
- Proxy サーバがメッセージをサブスクライブ



アーキテクチャ詳細

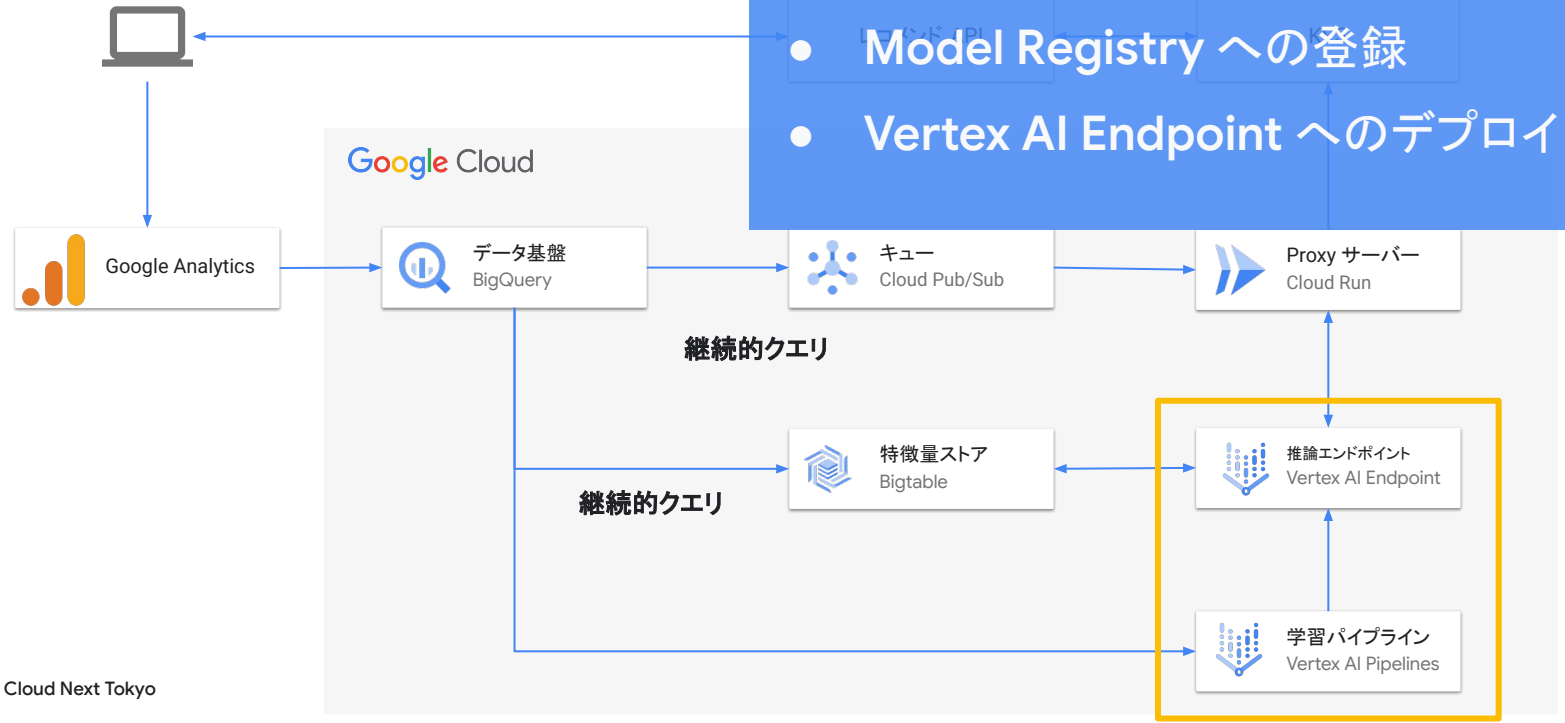
1. Proxy サーバから推論エンドポイント呼び出し
2. 推論エンドポイントはBigtableに保存された特徴量を用いて推論し、結果を返却
3. Proxy サーバから KVS に推論結果を書き込む



アーキテクチャ詳細

日次でパイプラインを実行

- Two-Tower モデルの学習
- Model Registry への登録
- Vertex AI Endpoint へのデプロイ



04. 得られた知見

うまくいったこと : ML/MLOps エンジニアの分業

責任分界点を明確にし、役割ごとの専門性にフォーカス
→ チームのパフォーマンスを最大化

MLOps エンジニア

役割:
システムの安定稼働と
開発の効率化

フォーカス:
再利用可能な共通基盤の構築

ML エンジニア

役割:
ビジネス価値の創出

フォーカス:
モデルの改善サイクル

うまくいったこと : ML/MLOps エンジニアの分業

MLOps エンジニア

API やパイプラインの安定稼働を保証し
横展開可能な共通基盤の開発に集中

- **アーキテクチャ設計**
  - 信頼性と拡張性、保守性を見据えたシステムの設計
- **共通基盤開発**
  - DMM の各サービスに展開しやすくするための共通ライブラリやCI/CD の整備
- **データ品質監視**
  - Dataplex 自動データ品質を利用した監視基盤の構築
  - データ起因の障害を未然に防止
- **SLO 監視**
  - システムの信頼性を定量的に評価・監視

うまくいったこと : ML/MLOps エンジニアの分業

ML エンジニア

ビジネス価値に直結するロジックの開発に集中

- **モデルの改善・実験**
  - 新規アルゴリズムの導入やロジック改善
- **モデルのオフライン評価**
  - ログデータを用いた定量評価
  - Streamlit を用いたレコメンド表示の可視化・定性評価
- **学習パイプライン・特徴量生成パイプラインの実装**
  - **BigQuery 継続的クエリを利用することで ML エンジニアだけで実装が完結**

うまくいったこと : 既存資産の活用

既に構築済みのシステム

- ユーザー行動をトラッキングし、BigQuery にイベントログをリアルタイム連携する基盤
- KVS に格納したレコメンド結果をユーザーに返却する API 基盤
- Vertex AI Pipelines による継続的なモデル学習基盤

新規実装部分

- レコメンドを更新するデータストリーム処理基盤

継続的クエリの採用により、シンプルかつスピーディに実装できた

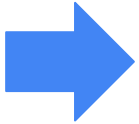
うまくいったこと : Bigtable のテーブル設計

- オンライン Feature store として Bigtable を初めて採用した
- Google Cloud 公式ドキュメントの設計ベスト プラクティスを MLOps チームメンバーで読み込み、スキーマ設計について議論を重ねた

-> 推論 API からの特徴量取得を低レイテンシで安定稼働させることができた

苦労したこと：継続的クエリのジョブ管理

- GA4 のログテーブルは日付シャーディングされたテーブル
- 継続的クエリの制約としてテーブルのワイルドカード指定ができない
- 日次で継続的クエリが参照するテーブルを切り替える必要がある



内製のジョブ管理システム構築

- 新しい日付のテーブル存在確認をするポーリング処理
- 古い継続的クエリを停止し、新しいクエリを開始

```
FROM
  `project.analytics_xxxxx.event_intraday_YYYYMMDD`
```

苦労したこと：Vertex AI Endpoint の日次更新

- Public endpoint の最大 RPS の制約やパフォーマンスの観点から Private endpoint を採用
- Private endpoint では、単一エンドポイントに同時に 2 つのモデルをデプロイし、トラフィックを切り替えることが不可能

Vertex AI Pipelines で日次実行する各サービス間で共通利用できる KFP コンポーネント を作成

1. 学習済みモデルを Model registry にアップロード
2. 新規 Private Endpoint を作成
3. モデルをエンドポイントへデプロイ
4. トラフィック切り替え - Proxy サーバ (Cloud Run) の環境変数を更新して推論API の URI を新エンドポイントに切り替え
5. 旧エンドポイントを削除

05. まとめ

まとめ

継続的クエリを利用したリアルタイム レコメンド システムを構築

1. ユーザーへのレコメンド反映を 1 日から 1 分に改善
2. ML エンジニアがビジネス ロジックに集中できる
3. MLOps エンジニアは安定稼働や横展開を見据えた共通部分の開発に専念