ホワイトペーパー

エンタープライズ向けに誕生し、クラウドでより強くなる

— Teradata VantageCloud Lake技術概要 —

Carrie Ballinger
Cloud Architecture Performance Solutions, Teradata

teradata.

目次

2 はじめに

3 多次元な並列処理機能

5 全方位のオプティマイザ

7 BYNETの多大な貢献

9 ワークフローの自律調整

11 ワークロード管理

12 ワークロード管理の進化

13 データベースの拡張性の進化

14 エコシステムにおける並列性の進化

15 クラウドネイティブな Terdata VantageCloud Lake

18 Teradata VantageCloud Lakeを
強化する既存機能

21 まとめ

はじめに

1930年代の大恐慌の時代、私の祖母は生き延びるために質素にならざるをえませんでした。つまり、何も無駄にせず、あるものを最大限に活用するということです。祖母の家では食べ物を捨てず、部屋を出るときは必ず電気を消し、小銭を節約しました。祖母が資源の乏しい時代に学んだのは、資源の効率的な利用だったのです。彼女は大恐慌時代に必要に迫られてこのような方法を採用しましたが、その特質は子孫に受け継がれました。何十年もの間、こうした倹約の習慣は、資源が豊富な時代になっても、私自身や私の家族に役立ってきました。

Teradataは、複雑なエンタープライズ向けプラットフォームを担ってきました。世界大恐慌をしのぐ祖母の家族のように、企業ユーザーは限られた柔軟性のないIT資産の世界で生きていく方法を学ばなければなりませんでした。Teradataは、設立当初、柔軟に拡張できない固定されたシステムサイズの中で顧客のニーズに応えるため、データベースの経済性を「おまけ」ではなく、必要なものとして、提供しました。

Teradataのアーキテクチャは、効率性とコストを最優先に考慮して設計されています。40年以上にわたってエンタープライズ分野での調整に重点的に取り組んできた結果、最小限の労力で高いパフォーマンスを実現できるようになりました。部屋を出るときに照明を消すように、無駄なリソースを排除したり、アルゴリズムを合理化することで、コストを削減しています。

Teradataがクラウドに進出した今でも、このアーキテクチャは生かされています。エンタープライズ向けの世界で生まれ、成長したTeradataは、成熟度、堅牢性、コスト効率、並列機能の深さ、最適化の多様性において高い優位性を誇っています。クラウドのリソースは無限かもしれませんが、予算は無限ではありません。Teradataは、エンタープライズ向けに特化し、磨き上げた職人技により、他のクラウドベンダーがその存在すら知らないような問題を解決し、クラウドでの地位を確立しています。

本ホワイトペーパーでは、Teradataデータベースの誕生当初からの主要な特性のいくつかを解説します。次に、当初の機能を強化した注目すべき進化のステップを取り上げます。最後に、Teradataが提供するクラウドネイティブサービス「Teradata VantageCloud Lake」において、これらの既存および進化したコンピテンシーが与える影響について考察します。

多次元な並列処理機能

Teradataデータベースは、SQL文の入力から実行の細部に至るまで、すべてが並列処理されるよう設計されています。このセクションでは、データベースに設計された3つの基本的な並列処理機能について説明します。

1. 全ての単位で並列処理

Teradataのデータベース内部の処理はすべて、AMPと呼ばれるあらかじめ定義された数の並列処理ユニットに分散されます。各AMPはデータベース内の小データベースのように機能し、所有するすべてのデータについての、ロード、読み込み、書き込み、ジャーナリング、リカバリなどを実行します。

AMPは並列処理の基本単位ですが、データベース内には、特にクエリパフォーマンス向上のために、さらに2つの並列処理機能：「ステップ内並列処理」と「マルチステップ並列処理」が組み込まれています。以下のセクションでは、これら2つの機能について説明します。

2. ステップ内並列処理

Teradataのオプティマイザは、クエリプランを作成する際、作業を「ステップ」と呼ばれる1つ以上の実行チャンクに分割します。各ステップには、複数の異なる処理を含めることができます。ステップ内並列処理とは、ステップ内の複数の処理が互いに流れ込み、複数の処理がオーバーラップして並列実行することです。（図2参照）

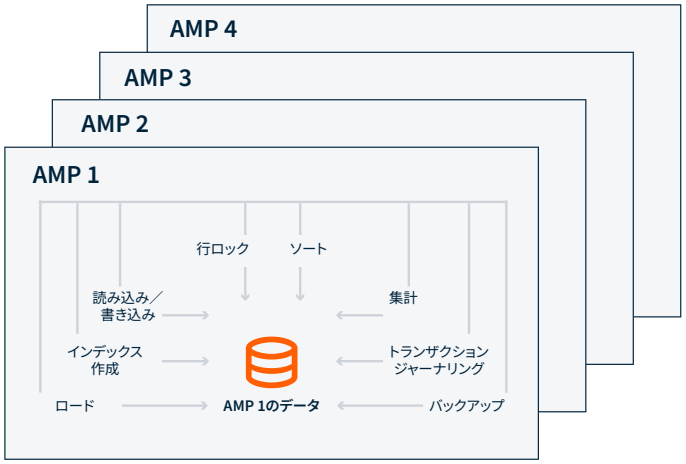


図1. 各AMPユニットの内部

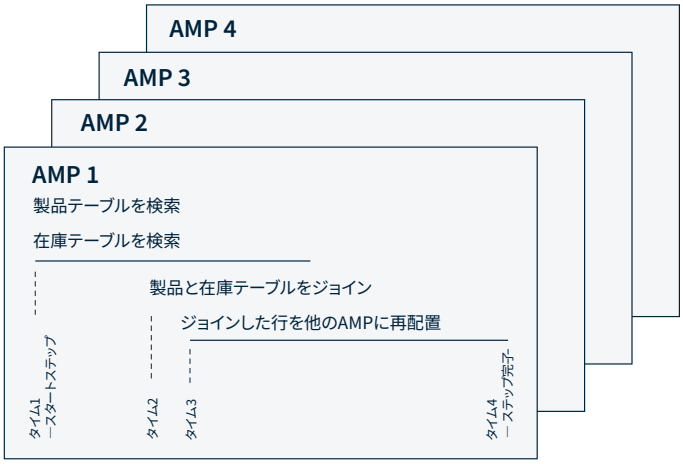


図2. 1つのクエリステップ内で並列に実行される複数のオペレーション

3. マルチステップ並列処理

オプティマイザが、クエリの複数のステップを、参加するすべての並列処理ユニットにわたって同時に実行することを選択した場合、マルチステップ並列処理が有効になります。（図3参照）

図3は、1つのクエリ実行をサポートする4つのAMPを示しています。さらに、クエリは7つのステップに最適化されています。ステップ1、2は、上記のセクションで説明したように、ステップ内並列処理性を示しています。ステップ1.1と1.2、2.1と2.2は同時に実行されるマルチステップ並列処理を示しています。

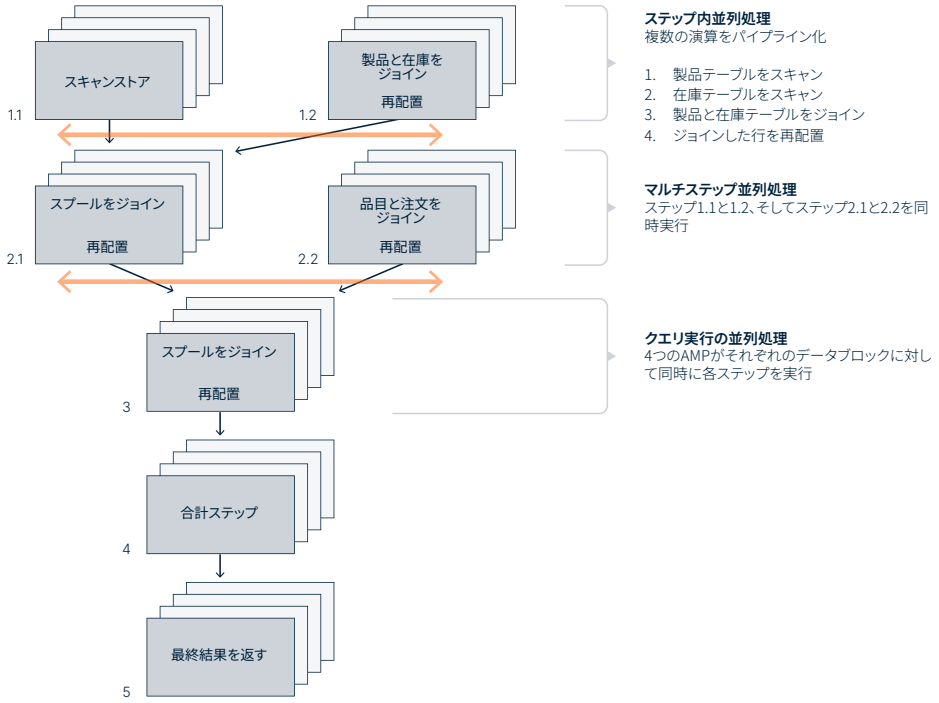


図3. マルチステップ並列処理

全方位のオプティマイザ

さまざまな並列技術は、各リクエストのニーズに合わせて慎重に適用されなければ、かえってパフォーマンスを損ねることになり、渋滞を招く可能性があります。異なる並列化技術のオーケストレーションは、「パーシングエンジン」（PE）と呼ばれるコンポーネント内に存在するクエリオプティマイザによって制御されます。

ジョインの実行計画

テーブルを直列的にジョインする（テーブル1とテーブル2を結合し、その結果をテーブル3に結合する、といった処理）ことは、特に何百ものテーブルをジョインするクエリでは、クエリ時間を増大させる可能性があります。

その代わりに、Teradataのオプティマイザは、より効率的なクエリプランを構築するために、異なるタイプのジョイン（ハッシュジョイン、マージジョイン、プロダクトジョインなど）を活用して、複数テーブルの同時ジョインを選択可能です。

下図は、6つのテーブルのジョインを最適化する際に、直列ジョインに制限されたプランと、ジョインステップの一部を並列に実行するオプションがあるプランの違いを示しています。

環境を評価

オプティマイザは、先述の並列処理に加えて、データ自体の特性、各ノードのAMP数、基礎となるハードウェアの処理能力など、多数の要因を考慮します。これらの情報をすべてまとめると、オプティマイザは推定コストを導き出します。複数のクエリプラン候補のそれぞれに使用されると予想されるリソースを計算し、最もコストの低い候補を選択します。

複数のクエリからのテーブルスキャンを同期

Teradataのプラットフォームのもう1つのコスト削減手法は、大規模テーブルの同期スキャンです。このTeradataのオプティマイザオプションは、別のセッションで進行中の同じテーブルのスキャンの現在位置で、新しいフルテーブルスキャンを開始することを許可します。テーブルスキャンを便乗することで、入出力（I/O）負荷が軽減され、より高い同時実行性がサポートされます。

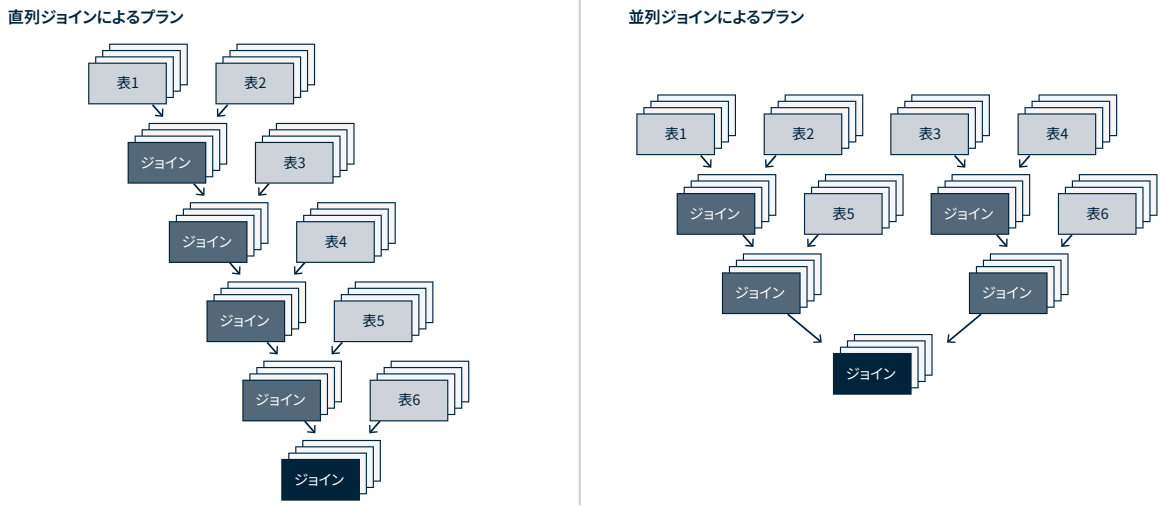


図4. 直列的なクエリプランと複雑なクエリプランの比較

オプティマイザの進化

基本的な部分は変わっていませんが、Teradataのオプティマイザは、変化する顧客のニーズに応えるため、時代とともに進化し続けています。

適応型オプティマイザ

通常、オプティマイザは静的なクエリプランを構築し、システムがそれを実行します。しかし、オプティマイザは単一のクエリ内で最適化と実行を交互に行うこともできます。これを適応型オプティマイザと呼びます。

適応型オプティマイザは、短いクエリや単純なクエリには適用されません。オプティマイザは、この手法を適用するタイミングを適切に選択し、あるしきい値（例えば、予想される実行時間）に基づいて、適用可能なリクエストを決定します。

適応型オプティマイザは、いわゆる「動的クエリプラン」を構築します。クエリを「フラグメント」と呼ばれるステップのブロックに分割します。最初のフラグメントに対してプランが構築され、AMP上で実行され、中間結果がオプティマイザに返されます。次に、前のフラグメントからの入力が考慮され、次のフラグメントが最適化され、実行されます。

クエリの書き換え

高度な最適化手法のもう一つの例としては、冗長なロジックを排除するためにクエリを書き換える方法があります。その一例として、データセットを何度も再構築するのではなく、計画の後半で複数のサブクエリへの入力として使用するために、クエリ内で一時的なデータセットを作成することが挙げられます。

クエリ書き換えの他の例としては、述語の簡素化、より最適なジョイン計画のためのビューの折りたたみ、および不要なカラムのプロジェクションを排除する「プロジェクションプッシュダウン」を達成するために、クエリ内のSQLコードのブロックを移動することが挙げられます。

リレーショナルデータと外部オブジェクトストアのジョイン

Teradataのオプティマイザは、Parquet、JSON、CSVなどの複数の異なるオープンファイルフォーマット（OFF）や、Apache Icebergなどのオープンテーブルフォーマット（OTF）を単一のクエリ実行に組み込むことができます。そして、その外部データをデータベース内のリレーショナルテーブルにジョインすることができます。

外部オブジェクトストアデータを使用したクエリの最適化の鍵となるのは、Teradataの並列処理アーキテクチャを活用するために、オブジェクトを構成内のすべてのAMPに均等に分散させることです。後の章では、初期の外部オブジェクト層であるOFFについて、この処理バランスがどのように実現できるか詳しく説明します。

オプティマイザは予算を握っているわけではありませんが、予算の使い道や予算がどこに割り当てられるかには大きな影響力を持っています。進化を続けるオプティマイザの機能の目標はすべて同じです。それは、Teradataのお客様が最高のパフォーマンスと、クエリごとのコストを可能な限り低く抑えられるようにすることです。

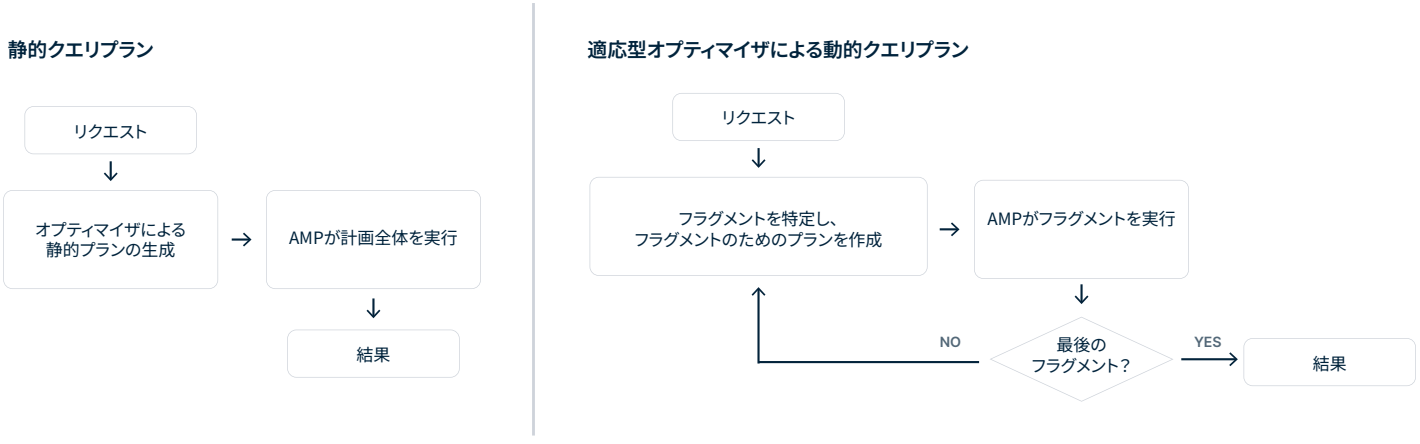


図5. 適応型オプティマイザによる動的なクエリプランの生成

BYNETの多大な貢献

Teradataアーキテクチャのもう一つの重要なコンポーネントは「BYNET」と呼ばれています。これは、すべての独立した並列処理コンポーネント間の相互接続として機能します。かつてオンプレミス型のTeradataシステムのハードウェア内に実装されていたこの機能は、現在ではソフトウェアとして実装されています。

BYNETは単にメッセージをやり取りするだけでなく、インテリジェンスとローレベルの機能を束ねたもので、クエリのライフサイクルのほぼすべての段階で効率的な処理を支援します。最適化されたクエリの各ステップの調整、監視、制御機能を提供します。

つまり、BYNETは航空管制官のような役割を果たし、システム全体が協調して動作し、異常な状況が発生した場合にそれを管理します。これには、ハードウェアの故障への対応、混雑箇所の監視、並列処理ユニット全体からの結果の順序付けなどが含まれます。

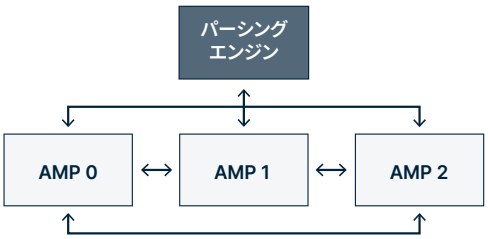


図6. AMPとパーシングエンジンはメッセージを使って通信

メッセージング

BYNETの重要な役割は、PEとAMP間の通信、およびAMPから他のAMPへの通信をサポートすることです。これらのシンプルなメッセージ転送要件は、より重厚な通信プロトコルをバイパスするローレベルのメッセージングアプローチを使用して満たされます：

- PEからAMPにステップを送信し、クエリステップを開始
- 異なるジョイン・ジオグラフィーをサポートするために、あるAMPから別のAMPに行を再配置
- 複数のAMPにまたがる（非常に大きな可能性もある）応答セットのソートとマージ

BYNETは、当初からエンジン内部のコスト削減がTeradataに浸透していることを示しています。メッセージプロトコルは低コストですが、Teradataは相互接続トラフィックを最小限に抑えることで、さらにコスト削減を実現しています。AMPの外部にデータを移動させることなく実行できるローカル処理が、可能な限り推奨されています。

AMP間のBYNET通信

BYNETが並列処理のすべての単位にわたって情報を統合・集約する能力を持っていなければ、各AMPは、進行中の各クエリステップについて同じ情報を得るために、システム内の他のAMPとそれぞれ個別に接続しなければならなりません。構成が大きくなるにつれ、クエリ作業を調整するこのような分散アプローチは、すぐにボトルネックとなってしまいます。

その代わりに、BYNETは同じクエリステップで動作しているAMP間に動的な関係を構築します。実行時のAMPの緩やかな関連付けは、各AMPのステップの完了や成功／失敗など、AMP間の通信に使用されます。これは、よりコストのかかるメッセージングではなく、セマフォを介したシグナリングによって実現されます。

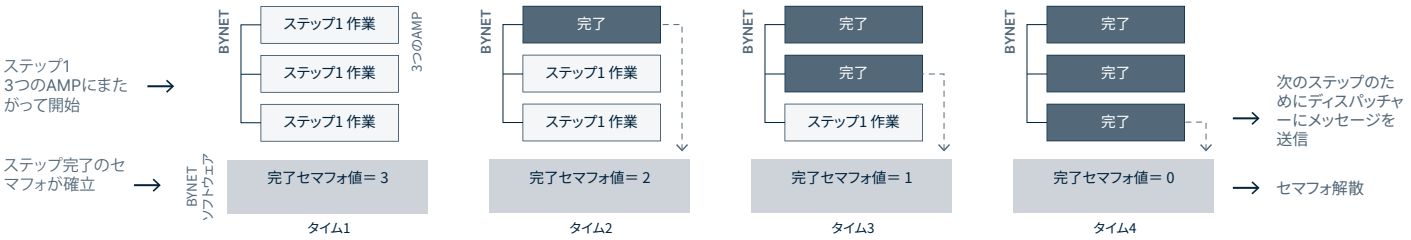


図7. 完了セマフォを使用したAMPの通信

最終応答セットのソートとマージ

クエリの最終応答セットをデータベース内に実体化する必要がないことは、以前からTeradataの差別化要因となっています。クエリの最終ソートとマージは、応答セットの行がクライアントに引き上げられる際にBYNET内で実行されます。この動的なI/Oなしのマージは、AMP、ノード、そして最終的にはPEレベルで同時に実行され、ソート順で最も高い値が、クライアントがさらに受信できる状態になるまで、クライアントバッファ内の応答セットに最初に格納されます。最終的な応答セットをまとめる必要がないため、リソースを大幅に節約できます。また、潜在的な「ビッグソート」のペナルティは排除されています。

これらの進化し続けるオプティマイザ機能の目標はすべて同じです。それは、Teradataのお客様が最高のパフォーマンスと可能な限り低いコストを享受できるようにすることです。

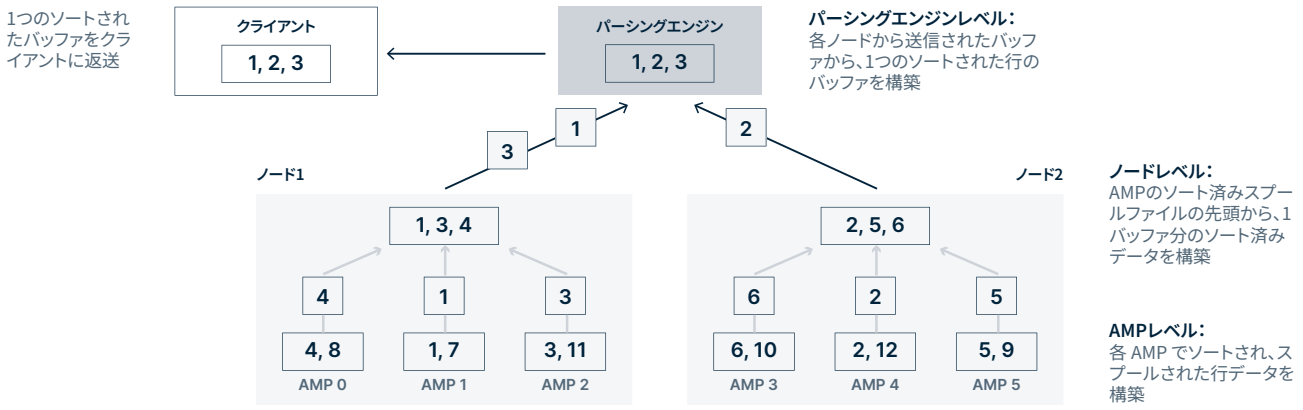


図8. BYNET内部での最終応答データの並べ替え

ワークフローの自律調整

シェアードナッシング並列処理型のデータベースでは、新しい作業をどれだけ受け入れられるか、また、1つまたは複数の並列処理ユニット内に引き起こされるかもしれない作業の渋滞をどのように特定するかを知ることは、重要な課題となります。その設計に内在するものとして、オプティマイザは、認識した各クエリに複数の並列処理を積極的に適用します。このアプローチは、各クエリのパフォーマンスとスループットのためにリソースを最大限活用します。しかし、システム全体のリソースを使い果たすことも容易に起こり得ます。Teradataのプラットフォームには、ワークフローを管理し、システムリソースを最適に活用するための数多くのテクニックが備わっています。

AMPレベルコントロール

Teradataのデータベースは、シェアードナッシングアーキテクチャに準拠し、システムに入ってくる作業の流れを高度に分散型で管理します。AMP間で、新規リクエストを保留すべきかどうかを判断するためのメッセージは送信されません。各AMPは、追加の作業を引き受けられるかを自己評価し、効率的に処理できる以上の負荷がかかった場合は一時的に処理を遅延させます。AMPが処理を一時停止した場合でも、その時間はごくわずかであり、多くの場合、ミリ秒単位のレベルです。

エンジン内部での作業の流れをボトムアップでコントロールすることで、データベースは、需要が非常に高くても低くても、即座に受け入れることができます。

AMPワーカータスク

AMPワーカータスク（AWT）は、各AMP内部でデータベースの作業を実行するタスクです。データベース起動時に各AMPに割り当てられたAWTは、バレットパーキング係員のように、作業が到着するのを待ち、作業を実行し、次の作業に移ります。

AWTはステートレスであるため、さまざまなデータベース実行のニーズに素早く対応します。各AMPには固定数のAWTが存在します。タスクを実行するには、利用可能なAWTを取得する必要があります。AMPあたりのAWT数に上限を設けることで、AMP内のデータベース作業のアクティビティ数を適正なレベルに保つことができます。AWTは、促進役と調整役の両方の役割を果たします。

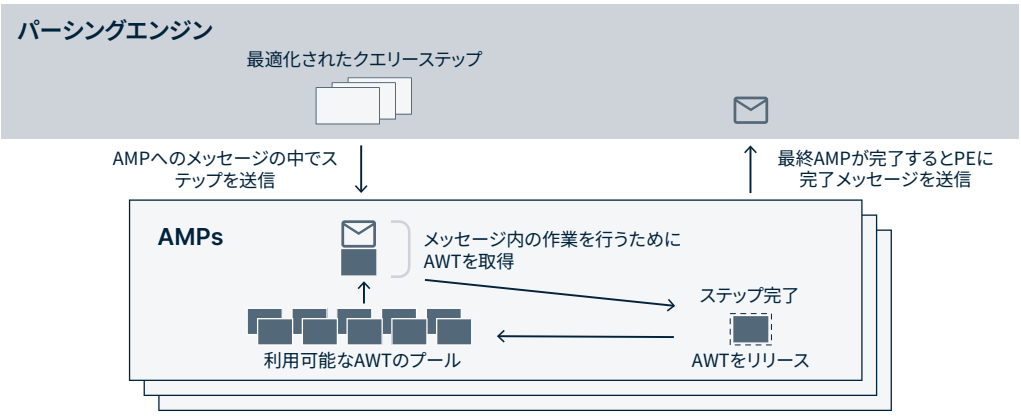


図9. AMP上でクエリステップを処理するAWT

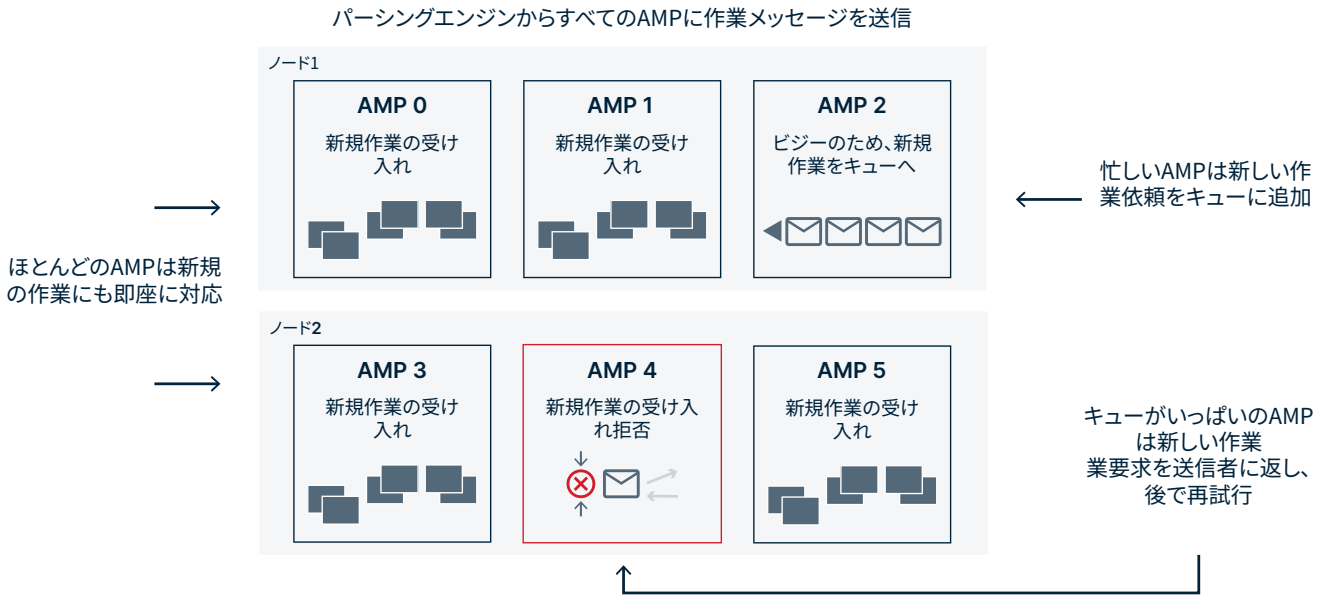
新規メッセージをキューに格納または拒否

AMP上のすべてのAMPワーカータスクが他のクエリステップの処理でビジー状態の場合、到着したワークメッセージはAMPのメモリ内に存在するメッセージキューに格納されます。メッセージが複数のAMPに送信される場合、一部のAMPは直ちにAWTを提供しても、他のAMPはメッセージをキューに入れる場合があります。これは、各AMPが独自のワークフローを管理する、ビジーなシステムにおける典型的な動作です。

メッセージキューが長くなりすぎると、到着したメッセージは拒否され、送信者に返送されて後で再送信されます。メッセージフローのオンとオフの影響はローカルに限定されます。つまり、その短い期間にメッセージが過剰に集中したAMPのみが一時的にスロットルバックするだけです。

フル活用の波に乗る

Teradataのプラットフォームはスループットエンジンとして設計されており、少数のクエリがアクティブな場合、並列処理を活用して各リクエストのリソース使用を最大限に高めることができる一方、需要の高い状況下でも応答セットを継続的に作成することができます。極端な使用状況下でもシステム全体の健全性を維持するために、このセクションで説明したように、高度に分散化された内部統制が基盤に組み込まれています。



10. 個々のAMPは作業が多すぎると一時的に押し戻す

ワークロード管理

前のセクションでは、Teradataデータベースへのクエリにて利用可能な多面的な並列処理に着目しました。また、オプティマイザがそれらの並列処理機能を賢く利用して、クエリごとにパフォーマンスを向上させる方法について説明しました。さらに、AMPレベルの内部制御により、ユーザーの高い要求レベルと過剰な並列性を管理する方法について説明しました。

PEレベルとAMPレベルでの自動制御に加えて、Teradataには常に、主に優先度の違いを制御するシステムレベルのワークロード管理が備わっており、内部データベースルーチンによって使用されています 。

当初設定された4つの優先順位

Teradataデータベースの初期設計者が直面した課題のひとつは、プラットフォーム上で最大限のユーザーリクエストをサポートしながら、必要に応じて内部データベースの重要なコードを迅速に実行する方法でした。例えば、トランザクションの中断によりロールバックが発生した場合、失敗をリカバリするための更新の取り消しを迅速に行うことができれば、システム全体にメリットをもたらします。

また、データベース内で実行中のバックグラウンドタスクが大幅に遅延しないようにすることも重要でした。これは例えるなら、市内の道路が自動車の交通渋滞で混雑し、救急車や消防車が数時間も通過できず、現場到着が大幅遅延するようなことがないように仕組みを整えることに酷似しています。

アーキテクトが発見したソリューションは、システム上で実行されるすべてのタスク（ユーザーが開始したもの、内部で実行されるもの）に優先順位を割り当てるシンプルな優先順位スキームでした。この基本的なアプローチでは、デフォルトを「中」に設定し、「緊急」、「高」、「中」、「低」の4つの優先順位が用意されました。

内部データベースルーチンやクエリコードの一部は、作業の重要度に応じて、他の優先順位のいずれかに割り当てることができます。例えば、「トランザクションの終了」ステップはすべて「緊急」の優先度に割り当てられています。これは、ほぼ完了した作業を高速で完了させることで、貴重なリソースをより早く新しい作業に割り当てることができ、データベース内の混雑を防ぐために重要であると考えられたためです。さらに、管理者があるユーザーに高い優先度を与えたい場合、優先度識別子の1つを手動でユーザーのアカウント文字列に追加するだけで済みます。

混合ワークロードの影響

Teradataユーザーが従来の意思決定支援クエリを、より多様化した新しいタイプのワークロードで補完し始めたことで、顧客の要件は時とともに変化してきました。

1990年代後半、一部のTeradataのユーザー企業では、標準的な意思決定支援クエリが実行されているのと同時に、在庫テーブルや顧客データベースなどのエンティティに対して直接参照クエリを発行し始めました。さらにコールセンターでは、Teradataデータベースで管理するデータを使用して、顧客アカウントとの最近のやり取りを参照するようにもなりました。オンラインアプリケーションが急成長したのも、バッチウィンドウを補完するために継続的なロードを採用するサイトが増え、エンドユーザーが最近のアクティビティに迅速にアクセスできるようになったのと同時期のことです。サービスレベルの目標がより重要視されるようになりました。現在では、Teradataのシステム上のクエリの80～90%が1秒以内に実行されるのが一般的です。したがって、より強力で柔軟なワークロード管理が必要とされています。

ワークロード管理の進化

作業の流れの内部管理はほとんど変わっていませんが、システムレベルのワークロード管理機能は、この25年間で劇的に拡大しました。最初の4つの優先順位を超える最初のステップとして、Teradataのエンジニアリング部門は、複数のリソースパーティションとパフォーマンスグループで構成されるより包括的な優先順位スケジューラを開発し、独自のカスタマイズされた重み付け値を割り当てる柔軟性を提供しました。これらのカスタム重み付けと追加の機能強化により、内部システム作業の制御を目的として設計された当初の機能よりも、ビジネスワークロードと優先順位の制御を容易に行えるようになりました。これは、固定サイズのエンタープライズプラットフォームにとって特に重要なことでした。

長年にわたって進化してきたワークロード管理の機能とオプションには、次のようなものがあります：

- サーバ名、アプリケーション、参照されるデータベースオブジェクト、オプティマイザの推定統計情報など、複数の分類による優先順位やその他の制御の割り当て機能
- 複数のレベルで配置でき、特定のタイプのクエリに適用できる同時実行制御メカニズム
- 記述が不適切なクエリや、特定の時間帯には実行に不適切なクエリを拒否するルール
- 現在の優先順位で消費されるリソースのしきい値を超える実行中クエリの優先順位を自動的に下げる機能

Teradataにおけるワークロード管理は、急速に拡大している分野であることが証明されており、Teradataのプラットフォーム上で多種多様な作業を組み合わせているお客様にとって不可欠なものです。

データベースの拡張性の進化

Teradataには、社内外で継続的にその能力を拡張してきた豊かな知見があります。カラムベースのテーブル、時系列データベース、地理空間データ型やテンポラルデータ型などの機能は、現在ではすべてTeradataのプラットフォームの一部となっています。強固なMPP基盤が、こうしたイノベーションを容易にしています。

Teradataが新たな方向性で成長し、コアコンピタンスを維持し続けられるのは、強固で実績のある基盤を築いてきたからに他なりません。

特殊関数の拡張

Teradataでは、スカラー関数、集約関数、テーブル関数、テーブル演算子、外部ストアドプロシージャなど、SQLの機能を拡張するユーザー定義関数（UDF）を作成することができます。この特殊関数のポートフォリオにより、エンドユーザーやTeradataパートナーは、JavaやC++でカスタムデータベースオブジェクトを作成し、Teradataのデータベースにこれまでなかった機能を追加することができます。これらの関数の一部は、データベース外でも動作します。

In-Database分析

Teradataは2000年代初頭に、上述の拡張機能を活用した並列処理でスケーラブルなIn-Database分析の先駆者となりました。その後、機械学習や人工知能（AI/ML）がTeradataの顧客ベースに広く浸透し、データ量がかつてないほどに増加するにつれ、この種の分析を行うためのローレベルの内部インターフェースを構築する必要が生じました。

これらの内部インターフェースの実装により、Teradataは、教師あり学習や教師なし学習を必要とするユースケースにおいて、従来考えられていた以上の拡張が可能になりました。これには、分類、回帰、セグメンテーション、時系列、デジタル信号処理などが含まれます。これらのローレベルのフレームワークは、同時に処理できる変数数や系列数、信号数に制限がありません。

また、これらの拡張機能により、従来のSQL以外の言語の処理も可能になりました。これにより、Teradataのプラットフォームの機能を活用した、まったく新しいユーザー像とユースケースが生まれています。

Bring Your Own Analytics（BYOA）

Teradataにとって、オープンな分析エコシステム全体におけるチームプレーヤーとなることは常に重要でした。例えば、TeradataのシェアードナッシングMPPアーキテクチャは、モデル推論やスコアリングに最適な手段です。推論に必要なすべてのデータが並列処理の単一ユニット上で利用できるため、戦術的な性質を持っています。

これにより、Teradataは、利用可能なモデル形式を作成できるあらゆる分析モデリングツールを受け入れることができます。H2O.ai、Dataiku、DataRobot、SASなどの特定の戦略的パートナーについては、Teradataのプラットフォームは、それらの実行環境にバインドすることで、それらのモデルを利用することができます。これらの技術のいずれかを使用することで、Teradataのプラットフォームは、多くの顧客のオープンな分析エコシステムにおいて、非常に並列性の高いスコアリングエンジンとなっています。

さらに、PythonとRのオープンソース言語とパッケージの人気が近年高まるにつれ、多くのPythonとRのスクリプトが分析パイプラインとして実稼働環境に導入されるようになりました。テーブル演算子を通じてこれらの言語を処理するTeradataの機能を基盤として、これらのパイプラインは最適化され、並列で実行することができます。SparkベースのPythonスクリプトでも同様です。

エコシステムにおける並列性の進化

この章では、Teradataの並列性をデータベースの外の世界に拡張するために行われたさらなる改善について説明します。

Native Object Store機能

今日の環境では、対象となるデータはTeradataのデータベース以外のファイルシステムやデータ管理プラットフォームに存在しているかもしれません。TeradataのNative Object Store機能により、クラウドベンダーのオープンファイルフォーマット（OFF）オブジェクトストアに格納されたデータをTeradata SQLを使用して直接読み書きすることができます。オプティマイザは、システム本来の並列性を最大限に活用するために、すべてのAMPにわたってCSV、Parquet、またはJSONファイルの読み取りと変換のタスクを割り当てます。

つまり、Teradataのデータベース以外のOFFのデータが利用できないということは決してありません。

QueryGrid™

QueryGridは、データアクセス、処理、およびテーブルレベルのデータ移動を1つ以上のデータソースに提供し、フェデレーテッドクエリを可能にするデータ分析ファブリックです。データソースは、同じ種類のもの（例えば、1つのTeradataプラットフォームが別のTeradataプラットフォームに接続する場合など）でも、異なる種類のもの（例えば、TeradataプラットフォームがGoogle BigQueryインスタンスなどのリモートサーバーに接続する場合など）でもかまいません。

QueryGridのユニークな機能のひとつは、2つのデータソース間で移動しなければならないデータの量を最小限に抑えるために、データに近い場所で述語のフィルタリングと処理を実行する機能です。QueryGridにより、お客様は以下が可能になります：

- データの移動を最小限に抑え、データを保存されている場所で処理
- データの重複を削減
- 分析処理とデータソース間のテーブルレベルのデータ移動を透過的に自動化

QueryGridは、Teradataのビルトイン並列処理を活用し、そのパフォーマンス能力を向上させるように設計されました。各AMPは、リモートサーバーからデータの一部を並列に取得して処理します。

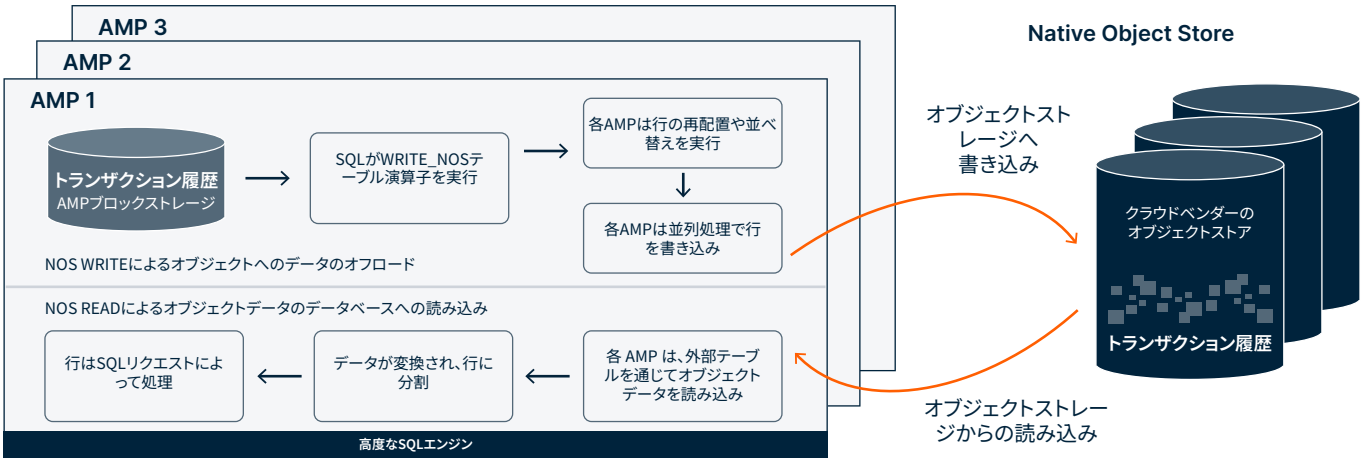


図11. Native Object Storeがオブジェクトファイルフォーマットのテーブルへのアクセスを並列処理

クラウドネイティブな Teradata VantageCloud Lake

TeradataのAI向けの包括的な機能を搭載するクラウドデータ分析プラットフォーム「Teradata VantageCloud Lake」は、エンタープライズの世界で確立された基盤をベースに構築されています。このセクションでは、Teradata VantageCloud Lakeの概要を説明し、特にそのアーキテクチャがクラウドの優れた機能を活用しながら、Teradataデータベースが持つ主要なメリットを維持している点について説明します。

プライマリクラスタとコンピュートクラスタ

Teradata VantageCloud Lakeのアーキテクチャは、TeradataをBYNET接続された固定のノード群で構成する単一の大型システムから、Teradata VantageCloud Lake環境を構成するビルディングブロックとなる、より小規模なクラスタの論理的な集合体へと移行させました。

Teradata VantageCloud Lake の各クラスタは、BYNET接続された個別の処理ユニットまたはノードのセットであり、各ノードには、他の従来のTeradataノードと同様に、おなじみのAMPとパーシングエンジンが含まれています。各クラスタは、他のクラスタとは独立して、AMP全体にわたって並列処理を行います。

プライマリクラスタは、すべてのTeradata VantageCloud Lake環境で必要です。プライマリクラスタには、ブロックファイルシステム（BFS）によって管理される永続的なブロックストレージにAMPがデータを所有する、常時稼働の固定ノードセットが含まれます。すべてのクエリは、プライマリクラスタで実行を開始し、終了します。クエリの解析と最適化、および最終応答セットの準備は、プライマリクラスタ上で実行されます。

2つ目のクラスタの種類は、コンピュートクラスタと呼ばれます。 コンピュートクラスタは、一時的な計算専用ノードの集合体であり、プライマリクラスタからクエリ作業の実行をリクエストすることができます。 コンピュートクラスタには、AMPとパーシングエンジンが含まれますが、プライマリクラスタとは異なり、ユーザーデータ用の永続的なブロックストレージは備えておらず、オブジェクトストレージからのデータアクセスに重点を置いています。

オートスケーリングと呼ばれる処理能力の自動追加または削除は、これらのコンピュートクラスタの間で行うことができます。 リソースの需要と使用状況に基づく内部的なしきい値が、処理能力の変更をトリガーします。

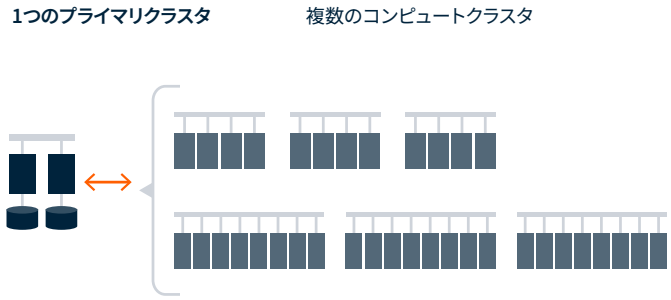


図 12. Teradata VantageCloud Lakeのアーキテクチャ

クラウドオブジェクトストレージ

Teradata VantageCloud Lakeは、コンピュートとストレージの分離をサポートしています。環境内のすべてのクラスタは、同じパブリッククラウドベンダーのオブジェクトストレージにアクセスできます。また、クラウドオブジェクトストレージはブロックストレージよりも低コストであるため、大量のビジネスデータをコスト効率よく保存および分析するために使用できます。しかし、ユニークな機能を持つBFSには依然として存在価値があり、必要に応じてプライマリクラスタはBFSを活用できます。

Teradata VantageCloud Lakeでは、複数の階層からなるオブジェクトストレージにアクセスできます：

- オブジェクトファイルシステム（OFS）ストレージ：Teradata VantageCloud Lake環境によって管理される独自のオブジェクトストレージで、ユーザーテーブル専用です。OFSに保存されたテーブルは、挿入、更新、削除、タイムトラベルをサポートしています。OFSは、効率的なアクセスをサポートするために、特別なインデックス技術で最適化されています。

- オープンファイルフォーマット（OFF）：OFF オブジェクトストレージは VantageCloud Lake 環境の外部にあり、セキュリティ規約によっては、他の複 数のプラットフォームまたは信頼できるサードパーティと共有される場合があります。データの定義は、Teradataのデータディクショナリ内に格納された「外部テーブル」内に保持されます。Native Object Store（NOS）機能を使用して、OFFデータを読み書きすることができます。

- オープンテーブルフォーマット（OTF）ストレージ：OTFに保存されたテーブルもTeradata VantageCloud Lakeの外部にあります。OTFは、オープンファイルフォーマットをサポートする、業界標準の「オープン」なACID準拠のテーブル形式です。OTFでは、テーブルのメタデータは一切Teradataのデータディクショナリに保存されません。その代わり、すべてのメタデータはオブジェクトストア自体に保存されます。

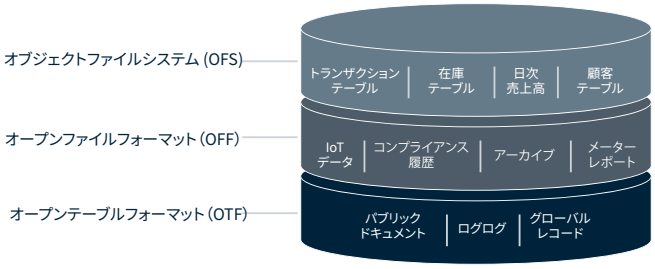


図13. Teradata VantageCloud Lakeからアクセス可能なオブジェクトストレージ層

Teradata VantageCloud Lake 向けオプティマイザの機能強化

Teradata VantageCloud Lakeに搭載されているオプティマイザには、Teradataのエンタープライズ向けオプティマイザで長年かけて完成させてきたすべての特別な技術と成熟度が備わっています。また、オプティマイザは、新しいTeradata VantageCloud Lakeアーキテクチャを認識し、その利点を活用するいくつかの新しい機能強化が含まれています。

グローバルプランナー

プライマリクラスタのオプティマイザに新たに追加された「グローバルプランナー」と呼ばれるコンポーネントは、プライマリクラスタまたはコンピュートクラスタ（コンピュートクラスタが利用可能な場合）でクエリプランの各ステップを実行する場所を決定します。 オブジェクトストレージからのデータの読み取りや処理、高度な分析などのリソース集約的な作業は、コンピュートクラスタに割り当てるよう最大限の努力が払われます。 Teradata VantageCloud Lakeのオプティマイザがクエリプランを構築する際には、各クラスタ内の処理能力のレベルを認識し、考慮します。

クエリステップ間のパイプライン処理

パイプライン処理は、Teradata VantageCloud Lake環境で利用可能なクエリ最適化機能拡張です。パイプライン処理により、各ステップ間の中間ファイルを排除することで、クエリの実行時間を短縮することができます。従来、クエリのプロデューサステップでは、条件を満たすすべての行を一時ファイルとしてディスクに書き込み、その後のコンシューマステップでそのファイルを読み込んでいました。パイプライン処理では、プロデューサステップで行を生成し、一時的なデータセットをディスクに書き込むことなく、メモリ内でその行をコンシューマステップに渡します。パイプライン処理では、データが生成されるとすぐに消費されます。

Teradata VantageCloud Lakeにおけるパイプライン処理は、オプティマイザが適用できるオプションです。さまざまなクエリ特性に基づいて、クエリ最適化の際に選択的に適用されます。これは数あるオプティマイザのオプションの1つであるため、Teradata VantageCloud Lakeのパイプライン処理は、元の最適化されたクエリプランに組み込まれた既存のニュアンスを維持します。Teradata VantageCloud Lakeのパイプライン処理の主な目的は、それが現実的で有益な場合に適用することであり、同時に最適化されたクエリプランを尊重することです。

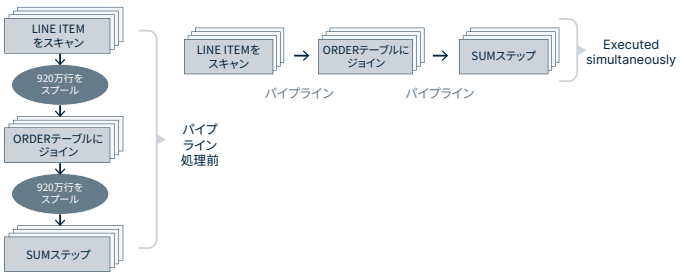


図14. Teradata VantageCloud Lakeのハイブリッドなアプローチによるクエリステップ間のパイプライン処理

マルチクラウド対応オプティマイザ（Amazon Web Services、Microsoft Azure、Google Cloud）

Teradataのプラットフォームでは、コストプロファイルは特定の構成の基本的なハードウェア能力に関する情報をオプティマイザに提供し、クエリプランの作成に影響を与えるために、進化してきました。この機能はTeradata VantageCloud Lakeで拡張され、Amazon Web Services（AWS）、Azure、Google Cloudなど、さまざまなプラットフォームおよびインスタンスタイプに関連するコストプロファイルが追加されました。

例えば、コストプロファイルには、特定のノードタイプのCPU MIPS評価やI/Oスループット評価などが含まれる場合があります。コストプロファイルを使用することで、オプティマイザは異なるベンダー構成を区別し、それぞれについて最適なプランを作成することができます。

1回のクエリでマルチストレージ接続

IoTデータをOTFテーブルから取得し、BFSストレージ内のキュレーションされた顧客テーブルとジョインし、その結果をOFFのトランザクションテーブルにジョインすることが、ビジネス上必要になる場合があります。これは、Teradata VantageCloud Lakeのオープンでコネクテッドなアーキテクチャとオプティマイザのインテリジェンスにより、単一のクエリとして実行できます。

Teradata VantageCloud Lakeのクエリオプティマイザは、データの格納位置と、各ストレージ層（BFS、OFS、OFF、OTF）に固有のパフォーマンス特性を認識しています。また、前述のセクションで述べたように、異なるベンダーの基盤ハードウェアの能力も認識しています。このような細部への配慮により、単一のクエリで異種データソースを最適な方法でジョインすることが可能になります。

Teradata VantageCloud Lakeのワークロード管理の自動化

Teradata VantageCloud Lake環境は複数のクラスタをサポートします。各クラスタは他のクラスタとは独立して動作します。個々のクラスタは他のすべてのクラスタから隔離されているため、固定システムのエンタープライズプラットフォームで必要とされる複雑なワークロード管理の要件は少なくて済みます。

そのため、各クラスタに自動的に設定される簡素化された優先順位のデフォルトのワークロード管理ルールセットが用意されています。クラスタ内で実行される作業に「明確な」優先順位の差を設けたい場合、管理者はこれらのデフォルトのワークロードのいずれかに特定のユーザーを割り当てることができます。

一方、明示的に優先順位が設定されていないユーザーから入力されたクエリは、自動優先順位付けの恩恵を受けます。クエリの実行予定時間に基づいて、5段階の「暗黙的」優先順位のうちの1つが選択されます。

実行が開始されると、暗黙的に優先されたクエリは、蓄積されたリソース消費量が指定の閾値を超えると、自動的に優先度が下げられます。

このシンプルで組み込み型のワークロード管理により、Teradataを初めて使用するユーザーでも、Teradata VantageCloud Lakeをうまく利用するためにワークロードに複雑なルールや条件を適用する必要はありません。また、経験豊富なユーザーは、ワークロード管理の決定事項の設定や監視から解放されます。

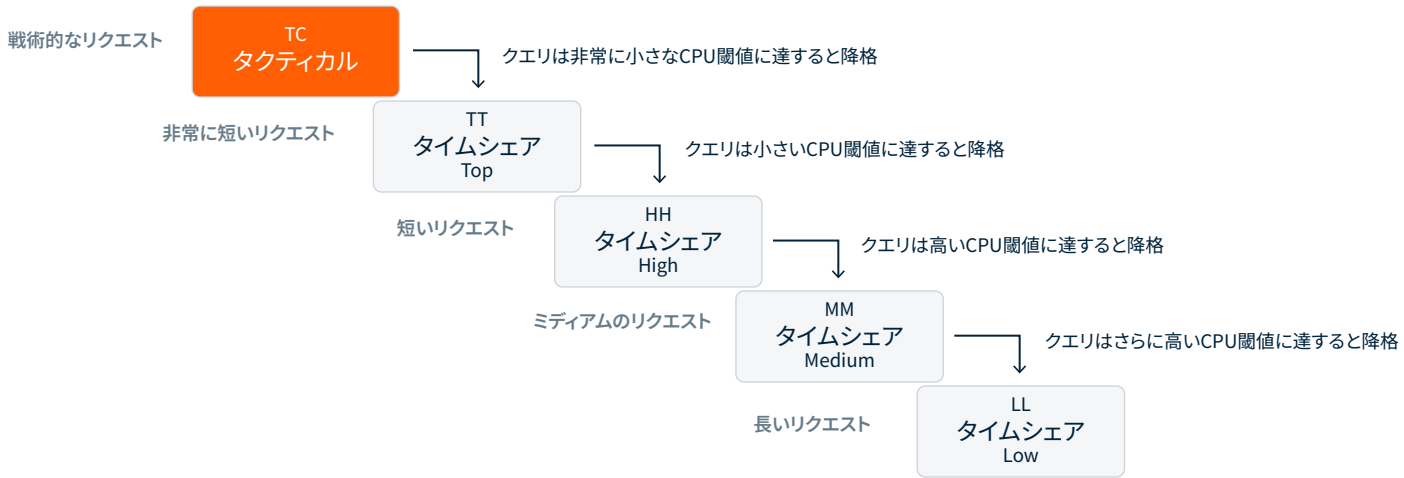


図15. Teradata VantageCloud Lakeでの自動優先順位付け

Teradata VantageCloud Lakeを強化する既存機能

このセクションでは、Teradata VantageCloud Lakeの機能を強化するTeradataの既存機能の一部について説明します。

QueryGrid™の進化

QueryGrid機能は、お客様のご要望に応じてTeradata VantageCloud Lake上にプロビジョニングされます。ノードへのインストールは、QueryGridプロビジョニングの一部として自動的に処理されます。Teradataのエンタープライズプラットフォームまたは Teradata以外のシステムにおけるQueryGridセットアップには、Teradata VantageCloud Lakeコンソールを通じて提供されるセルフサービスガイドに従っていただく必要があります。

Hive、Spark、および汎用JDBCコネクタは現在、AWSとAzureのTeradata VantageCloud Lakeでサポートされています。BigQueryはまもなく利用可能となり、Hive/Sparkもその後すぐに利用可能となる予定です。今後、QueryGridコネクタがさらに追加され、他のデータソースへのアクセスをさらに拡大します。各環境は、オンプレミスプラットフォームやCSP内のその他のサポート対象プラットフォームに接続できるため、拡張されたQueryGridにより、真のハイブリッドマルチクラウドソリューションが実現します。

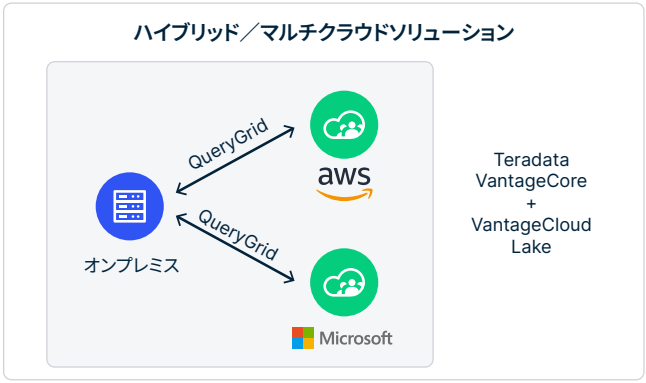


図 16. QueryGridが実現するハイブリッド／マルチクラウドソリューション

さらに重要なこととして、QueryGridテクノロジーは、ユーザーには透過的に見える形でプライマリクラスタとコンピュートクラスタを接続する経路として、Teradata VantageCloud Lakeのインフラに組み込まれています。この内部通信レイヤーは、異なるクラスタ間のデータ、メタデータ、最適化されたクエリステップの受け渡しをサポートし、Teradata VantageCloud Lake以前から長年にわたって存在する定評のあるQueryGrid機能に依存しています。

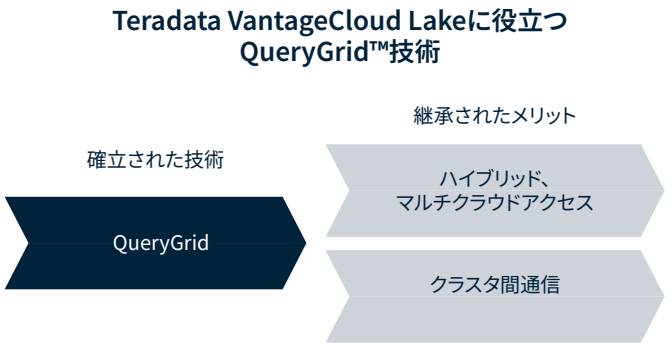


図17. Teradata VantageCloud Lakeで活用するQueryGrid技術

OTFを活用するための既存技術の進化

OTFにより、オープンで標準的な方法でデータを保存し、複数のコピーの巨大なデータセットを管理する必要なく、異なるコンピューティング、分析エンジン、ツール間で簡単に共有することができます。Teradataは、マルチクラウドおよびマルチデータレイク環境において、さまざまなOTFおよびオープンカタログをサポートしています。AWS Glue、Hive、Unityなどの複数のオープンカタログをサポートすることで、メタデータの複製が不要になり、相互運用性が促進されます。IcebergとDelta Lakeは、Teradata VantageCloud Lakeで最初にサポートされた2つのOTFです。

Teradata VantageCloud Lake環境からのOTFテーブルへのアクセスは、Native Object Storeを使用してOFFテーブルにアクセスするのと同様の方法で実装されています。クラスタ内の各AMPは、OTFテーブルのオブジェクトを読み取るための一部のシェアが割り当てられます。そして、ネイティブオブジェクトストレージへのアクセスと同様に、すべての操作と変換が自動的に並列処理されます。オプティマイザは、OTF内のデータにアクセスするすべてのステップをコンピュートクラスタに送信し、クラスタの処理能力と並列処理能力に基づいてプランを構築します。

OTFとTeradata VantageCloud Lakeでサポートされているその他のストレージ層との違いの1つは、OTFテーブルのメタデータはTeradataのデータディクショナリではなく、オブジェクトストア自体に保存されることです。メタデータとのやり取りのフォーマット、構成、プロトコルは、Iceberg、DeltaLakeの仕様内で定義されており、このアプローチにより、他のOTFベンダーもOTFデータセットに同時にアクセスし、操作することが可能になります。ただし、セキュリティ上の理由から、OTFデータへのアクセスにはすべて権限の付与が必要です。

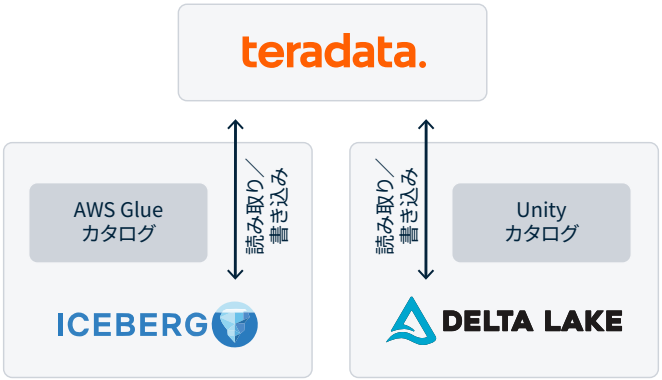


図18. IcebergとDelta LakeにおけるOTFデータへのアクセス

OTFテーブルとカタログは、ユーザーが所有・管理するクラウドストレージに保存されます。Teradata OTFは、クラウドプロバイダーの認証および承認メカニズムと統合され、OTFテーブルとメタデータへのアクセス制御を遵守します。さらに、ネットワーク接続は安全に保護され、転送中および保存中のすべてのデータは暗号化されます。

TeradataのシングルAMPアクセス最適化は、スキーマの検出や特定のカタログ内の現在のメタデータへのアクセスにおいて、OTFテーブル情報を取得するための非常に効率的な方法を提供します。しかし、OTFデータ自体へのアクセスとなると、各AMPがOTFデータの読み取りと変換作業を分担することになります。何年にもわたって複雑なクエリのパフォーマンスを可能にしてきた同じオプティマイザがOTFテーブルをファミリーの一員としたため、OTFテーブルは、あらゆるストレージ層から取得した他のリレーショナルデータや非リレーショナルデータと容易に結合できるようになりました。

OTFの読み取りや書き込みなど、Teradataのプラットフォーム上で実行されるすべての処理は、ワークロード管理と緊密に統合されています。このような常時使用されるワークロード管理技術により、管理者はOTFの読み取りや書き込みを任意の優先順位で実行するように割り当て、同時実行を効果的に制御することができます。

Teradata VantageCloud Lakeの
オープンテーブルフォーマット実装のメリットとなるテクニック

確立された技術	継承されたメリット
AMP全体で並列アクセス	各AMPがOTFテーブルを並列に読み取ることでパフォーマンスを向上
高度なクエリ最適化	クエリ計画メカニズムにより、OTFやブロックまたはオブジェクトストレージ内の他のテーブルとの効率的なジョイン計画を生成
Native Object Storeインフラストラクチャ	メタデータの読み取り、行のフィルタリング、および読み取ったデータのリレーショナルデータベース形式への変換に同じインフラを使用

図19. 確立された技術を基盤としているOTF実装

Teradata VantageCloud Lake におけるアナリティクスの進化

先に述べた拡張メカニズム（ユーザー定義関数、外部ストアドプロシージャ、In-Database分析、オープンソース言語のサポート）は、現在のTeradata VantageCloud Lakeでは非常に有用です。主な利点は、これらの負荷の高いアプリケーションを独自のコンピュートクラスタに分離し、専用のリソースを活用して他のアクティブな作業に干渉しないようにできることです。

Teradata VantageCloud Lakeは、アナリティクスに関連するさらに重要な機能が追加されました。それが、専門用途向けコンピュートクラスタです。専門用途向けコンピュートクラスタは、複雑でリソース集約的なアナリティクスアプリケーションのニーズを満たすために特別に設計されています。

例えば、アナリティクスクラスタは、TeradataのOpen Analytics Frameworkを使用して、コンテナ内でテーブルデータに対してユーザースクリプトを実行するための専用クラスタです。アナリティクスコンピュートクラスタが異なるのは、専用ハードウェアが搭載されており、その上で実行されるステートメントには、より多くのメモリとCPUが割り当てられている点です。

2つ目のタイプの専門用途向けコンピュートクラスタは、グラフィックス処理ユニット（GPU）クラスタです。Teradata VantageCloud Lake環境では、専門用途向けGPUユニットをプロビジョニングし、複雑な数学的計算を効率的かつ並列的に実行するために活用することができます。

AIの進化

生成AIは新しい概念ではありませんが、OpenAIがChatGPTを立ち上げた2022年後半以降、この技術が注目されるようになりました。

Teradata VantageCloud Lakeは、AIが主流になりつつあった時期とほぼ同時期に登場しました。大規模言語モデル（LLM）をサポートする環境を提供し、また、パフォーマンスとスケーラビリティを向上させるために並列処理できるという利点により、生成AIをサポートしています。Teradataの拡張機能とTeradata VantageCloud Lakeの専門用途向けコンピュートクラスタにより、これが実現します。推論とモデルのファインチューニングの両方において、顧客の実際のデータを使用して、同じデータが安全ではない可能性があるパブリックLLMに頼ることなく、Bring Your Own LLMが実現可能です。

Teradata VantageCloud Lakeは、Amazon BedrockやAzure OpenAIServiceなどの他ベンダーのLLMとも統合できます。これは、インテリジェントなドキュメント検索機能であるTeradataのask.aiが採用しているアプローチで、自然言語クエリにも進化させることができます。例えば、ask.aiは、地域別の今月の収益合計を計算するといった詳細なリクエストに対して、正しいSQL構文を提供し、その結果を意味のある形で提示することができます。

Teradataのデータベースの拡張性、In-Database分析、AIといった分野における進化は、すべて今日のTeradata VantageCloud Lakeの提供という形で結実しました。この新しいクラウドアーキテクチャは、これらの以前の機能を強化し、Teradataのお客様が1〜2年前には考えられなかったようなことを可能にしています。

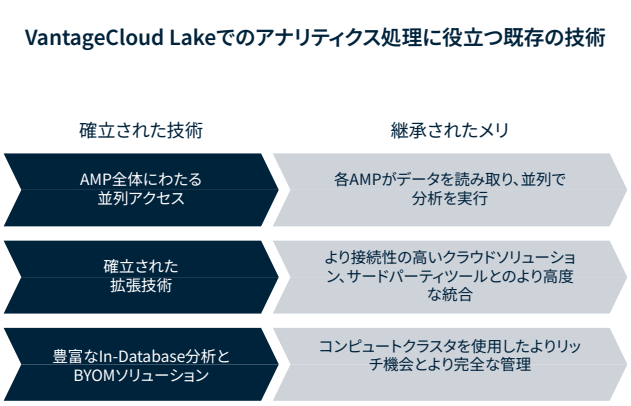


図 20. Teradata VantageCloud Lakeのアナリティクスは、確立された技術により強化

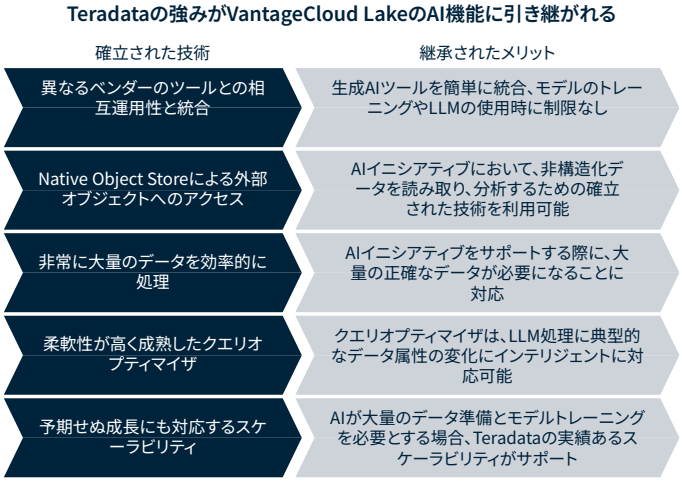


図21. Teradata VantageCloud LakeのAI機能も確立された技術により強化

まとめ

基礎は重要です。Teradataが新たな方向性を見出し、コアコンピテンシーを維持し続けているのは、強固で実績のある基礎を築いてきた結果です。Teradataのデータベース技術が成長し成熟するにつれ、同じ基本原則が新たな技術革新にも適用されてきました。これは特にTeradata VantageCloud Lakeに当てはまります。

Teradataのデータベースを構成する重要な基本機能について理解を深めることで、そのアーキテクチャの優雅さと耐久性を実感することができます。これらの機能は、多くの点で一貫性を保っています。同時に、多くの機能がそのオリジナルのアーキテクチャから進化を遂げています。つまり、置き換えるのではなく、それを基盤として構築されているのです。

これらの基盤コンポーネントは、Teradataが前進し続ける中で、メインフレームデータベースとの統合、混合ワークロードの管理、分析の世界の進歩など、あらゆる場面で不可欠であることが証明されています。エンタープライズで生まれたということは、クラウドでも強みを発揮することを確かに予見しています。

Teradataについて

Teradataは、より良い情報が人と企業を成長させると信じています。Teradataは、AIのための最も包括的な機能を搭載するクラウドデータ分析プラットフォームを提供します。統合された信頼できるデータと信頼できるAIを提供することで、より確信に満ちた意思決定を可能にし、より迅速なイノベーションを実現し、企業が最も必要とするインパクトのあるビジネス成果の獲得を支援します。詳細は**Teradata.jp**をご覧ください。

17095 Via Del Campo, San Diego, CA 92127　Teradata.com

teradata.