

# SNOWFLAKEによる データエンジニアリング:

データのスムーズな取り込み・変換・提供により  
リアルタイムインサイトを実現

## 高性能のデータパイプラインはデータドリブン型組織の基盤

データパイプラインは、今日の企業運営に不可欠な要素です。有効なデータパイプラインを備えているかどうかは、データアーキテクチャが事業に真の価値をもたらすか、足かせになるかの分かれ目となります。

企業は、24時間、365日、ウェブや企業のアプリケーション、モバイル、IoTデバイスからデータを受信しています。データパイプラインは、データサイエンティストやアナリストがデータを分析し、ダウンストリームのアプリケーションがその後の利用に向けてデータを処理している間に、データをロードし処理する必要があります。

データエンジニアは、データを収集し、変換し、さまざまなビジネスラインに提供しつつ、最新の技術革新に対応してビジネスの要求を先取りする必要があります。しかし従来のレガシーアーキテクチャではあらゆる段階で問題が発生するため、データエンジニアのタスクがより複雑なものとなってしまいます。

## レガシーアーキテクチャがデータパイプラインに及ぼす影響

従来のデータアーキテクチャは、最新のアナリティクスや効率的なデータエンジニアリングの要件を満たすようには構築されていません。図1は多くの企業で見られる典型的なデータパイプラインを示しています。これは、生データ源から必要な提供先へのデータ配布のために構築された複雑なテクノロジーの網目です。

統合、変換、集約、提供のそれぞれの段階で組織は複数のツールを展開しており、これらすべての接続ポイントが障害点となる可能性があります。こうしたアプローチではガバナンス、セキュリティ、サイロ化の問題が頻繁に発生します。これは、ビジネスのさまざまな領域における多様なニーズを満たすために多くのデータコピーが必要となるためです。その結果、組織はインフラストラクチャーの管理に多くの時間を費やすことになり、真にデータを有効活用できる時間が少なくなります。

「データの多様化が進み、適切なデータを適切な人に適切なタイミングで提供する必要からデータエンジニアリングのニーズが生まれました。データやアナリティクスのリーダーは、データエンジニアリングをデータ管理戦略に統合する必要があります。」

### Gartner Research

「データエンジニアリングはデータとアナリティクスの成功に不可欠」、  
2019年12月18日<sup>1</sup>

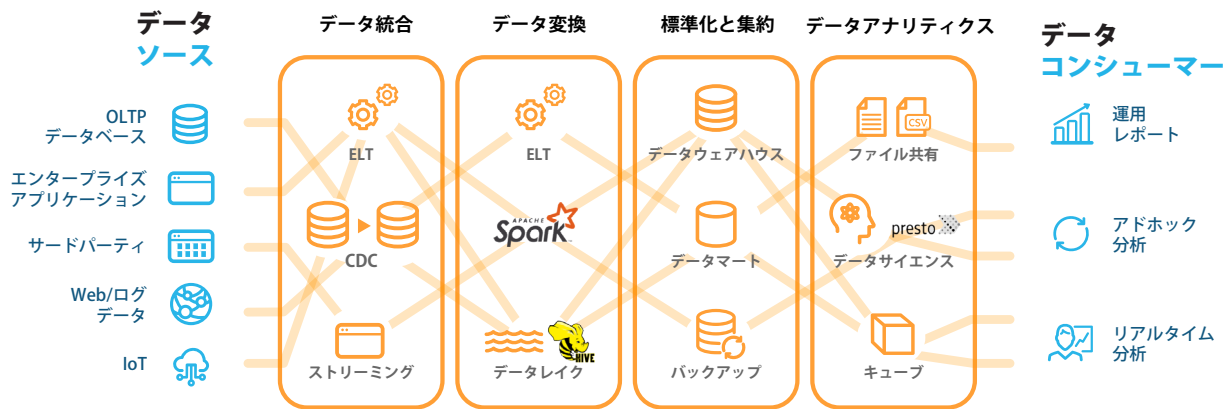


図1:従来のデータアーキテクチャは複雑でコストがかかり、制約を伴います

従来のレガシーアーキテクチャは、データエンジニアにとって図2に示す以下の課題の発生源となります。

- サイロ化された多様なデータがあり、統合されるスピードもまちまちである
- リソースの競合により性能や信頼性が損なわれる
- 複雑なパイプラインとアーキテクチャを構築して管理するには、膨大な時間と多彩なスキルセットが必要となる

### サイロ化した多様なデータ

レガシーアーキテクチャでは、多くの場合、さまざまなデータベース、データレイク、データマート、さらにはクラウドやオンプレミスにデータが存在します。これらのデータはそれぞれ複数の異なる形式で作成され、その生成ス

ピードもさまざまです。このような複雑さに加えて、アプリケーションによって、想定されるデータの提供方法も異なっています。新しいデータがすぐに利用可能にならないとアプリケーションやアナリストは必然的に古いデータを使用せざるを得ないため、結果的に意思決定者が重要なインサイトを見落とすことになり、十分に事業状況を可視化することができなくなります。

### 信頼性と性能の低下

ただでさえ複雑なデータパイプライン上でワークロードが増すことで、信頼性が低下し、データの変換に遅れが生じます。リソース不足によりパイプラインが意図したとおりに機能しない場合、データ利用者に不利益が生じないようパイプラインを修正せねばならず、データエンジニ



図2: データエンジニアはアーキテクチャとパイプラインの課題に苦しんでいます

アに多大な負荷がかかります。ロード専用のウィンドウでは問題は解決しません。遅くて不便なパイプラインや古いデータの蓄積の原因となるだけです。

### 複雑なパイプラインとアーキテクチャ

多くの場合、データエンジニアリングにはカスタムコーディングやさまざまなツール、多彩なスキルセットが必要です。複雑なパイプラインやデータサービスは、システムをつなぎ合わせて構築されているため、構成や管理対象となるインフラストラクチャーが増え、パイプラインの構築や維持の複雑さやコストが増大することになります。キャパシティプランニング、性能の微調整、同時実行により、パイプラインの維持はさらに複雑なものとなり、その結果データエンジニアは、ビジネスSLAを満たすデータの提供よりも、複雑なパイプラインの構築、APIの管理、インフラストラクチャーの維持に多くの時間を費やすようになります。

### SNOWFLAKEは、データエンジニアによる最新のインサイト提供を支援

Snowflakeは、データエンジニアリングの合理化と優れたパフォーマンスと利便性の実現により、データ提供用パイプラインやインフラストラクチャーの管理に追われることなくデータから多くの価値を引き出すことを可能にします。Snowflakeのマルチクラスター型共有データアーキテクチャは、データの処理やロード、変換、アナリティクス向けにそれぞれ独立した複数のクラスターを割り当てることができます。同じデータの同時共有が可能なためリソースの競合もありません。データの移動や変換を伴わずに、今ある場所でそのままデータの処理が可能です。

図3に示すように、Snowflakeを利用したデータエンジニアリングには以下のメリットがあります。

- すべてのデータを一元化されたプラットフォームにストリーミングまたはバルク統合
- すばやい拡張性を備え、リソース競合のないデータパイプラインを運用
- 合理化されたアーキテクチャと拡張性のあるデータパイプラインで生産性を強化

Snowflakeがデータエンジニアリングに最適な理由は次のとおりです。

- 構造化データと半構造化データの両方に対応(まもなく非構造化データにも対応予定)
- 一括かほぼリアルタイムでデータを取り込み
- ロード時のエラーに自動的に対処し、データの重複を排除
- パワフルでベストオブブリードな変換エンジンを使用して、Snowflake上で直接データを処理
- SQLと拡張性(機能やストアプロシージャなど)の活用により、データパイプラインとアーキテクチャを合理化
- ライブデータシェアリングを通じて、簡単で安全なコラボレーションとユーザーへのデータ提供を実現
- 一般的なドライバ(ODBC、JDBC、Python、Spark)のサポートが組み込まれており、お使いのデータレイクにエクスポート可能

データの取り込み、変換、提供にSnowflakeをどのように利用するかについては、以降のセクションを参照してください。

<p>あらゆるスピード、すべてのデータ</p>  <p>すべてのデータを単一のプラットフォームでストリーミングまたはバッチで統合</p>	<p>パイプラインの性能と信頼性の向上</p>  <p>すばやい拡張性を備えつつリソース競合のないデータパイプラインの運用</p>	<p>パイプラインの複雑性の緩和</p>  <p>合理化されたアーキテクチャと拡張性のあるデータパイプラインによる生産性の強化</p>
---	--	--

図3: Snowflakeによりデータエンジニアリングが合理化され、真に意義のあるデータ利用に注力できるようになります

## 自由なスピードで、すべてのデータを取り込み

Snowflakeのデータパイプラインは、バルクまたは継続のいずれかを選択でき、Snowflake上で直接データを処理します。Snowflakeではマルチクラスターのコンピューティングアプローチを採用しているため、これらのパイプラインはその他のワークロードの性能に影響を与えることなく複雑な変換に対応できます。すべてのデータをスト

リーミングまたはバッチで簡単に取り込み、安全に一元管理できるため、ユーザーは構造化データと半構造化データの両方をネイティブ状態で利用できます。

下表と次セクション以降で説明するように、Snowflakeは複数のデータ取り込み方法を提供します。

方法	目的
バルクロード	クラウドストレージにすでにあるファイルからデータのバッチをロードするか、ローカルマシンからSnowflakeにデータファイルをコピーし、COPYコマンドを使ってデータをテーブルにロードします
Snowpipeによる継続的ロード	少量ずつデータ(マイクロバッチ)をロードし、段階的にアナリティクス向けに利用可能にします
Kafka用Snowflakeコネクタ	Apache Kafkaからイベントストリームを取り込み、Snowflakeテーブルにロードします
Snowflakeデータマーケットプレイス	他者が公開したクエリ対応ライブデータを取り込みます

以下の図4では、Snowflakeのバルク処理、継続的処理、Apache Kafkaデータロードの仕組みを比較しています。

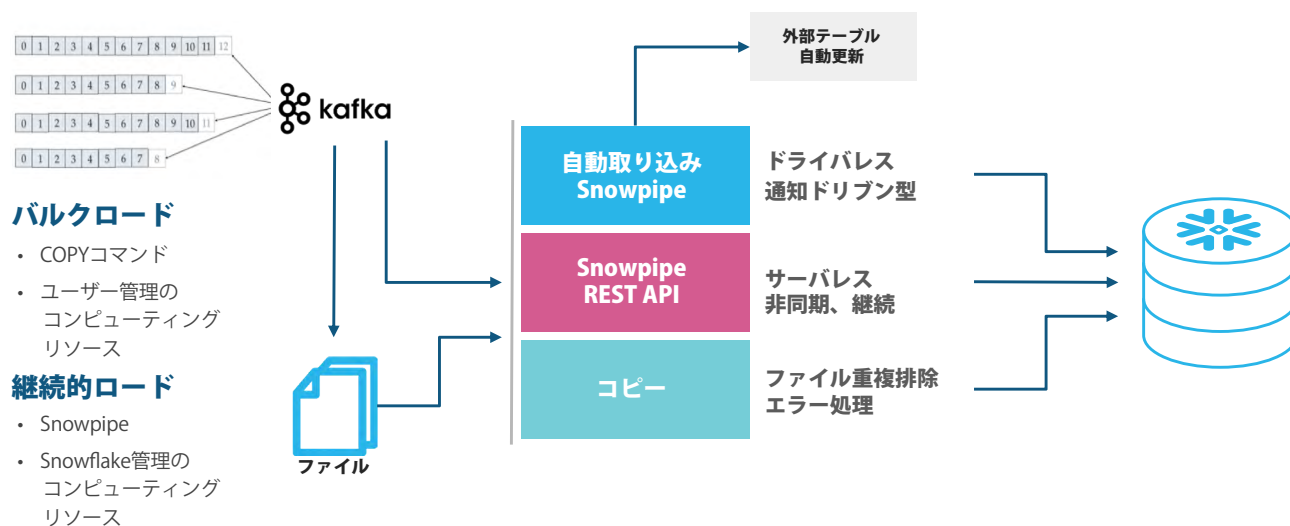


図4: Snowflakeを採用すると、データエンジニアは非常に柔軟な方法でデータをロードできるようになります

### COPYコマンドを使用したデータのバルクロード

ロードするデータ量とデータロードの頻度に応じて、COPYコマンドによるバルクロードを使うべきかどうかを判断しましょう。バルクロードオプションを利用すると、クラウドストレージにすでにあるファイルからデータのバッチをロードするか、ローカルマシンから内部のSnowflakeクラウドストレージのロケーションにデータファイルをコピー（ステージ）し、その後COPYコマンドを使ってデータをテーブルにロードすることができます。

バルクロードは、ユーザー提供の仮想ウェアハウスに依存します。この仮想ウェアハウスは、COPY命令内でユーザーが指定します。予測されるロードに対応できるようにウェアハウスのサイズを適切に設定する必要があります。

Snowflakeは、COPYコマンドを使用してデータをテーブルにロードする際のシンプルな変換をサポートします。

オプション:

- 列の並べ替え
- 列の省略
- キャスト
- ターゲット列の長さを超えるテキスト文字列の切り捨て

### Snowpipeを使用した継続的データロード

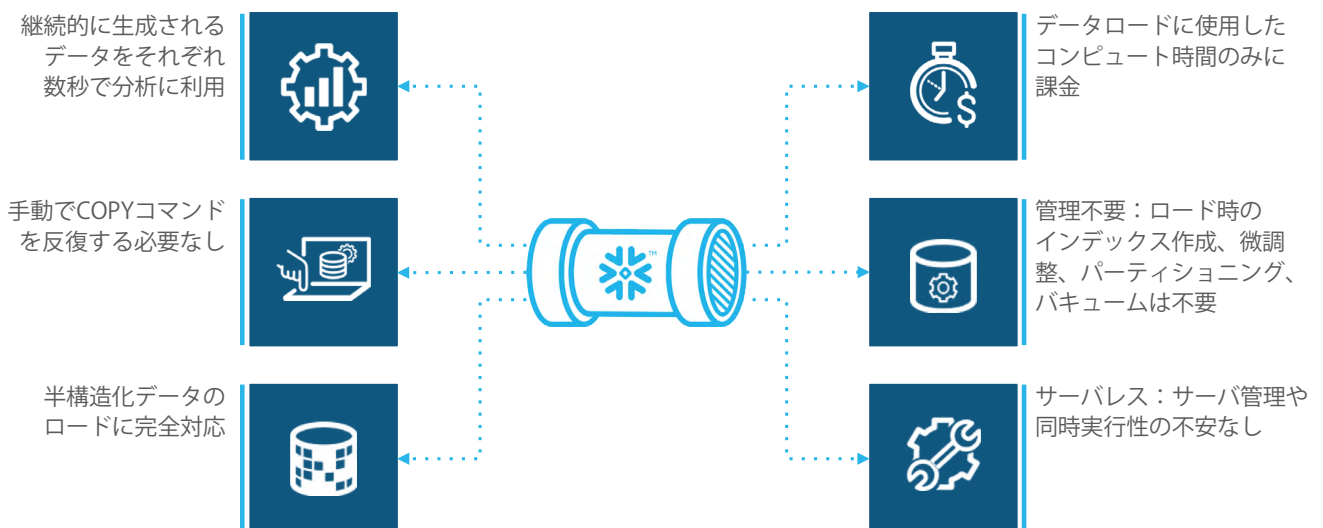
データアーキテクチャの進化に伴い、データウェアハウス内でのデータのロードや処理方法に関する要件も進化しています。データドリブン型の企業が業務を合理化し、顧客へのサービスを向上させ、新たな市場機会を発見する

ためには、ライブデータへの迅速かつ瞬時のアクセス性と瞬時のアナリティクス性能が不可欠です。

Snowpipeは、ステージ内で利用可能となったファイルからすぐにデータを自動的にロードできるサーバ不要のツールです。少量ずつデータ（マイクロバッチ）をロードし段階的にアナリティクス向けに利用可能にしていくSnowpipeは、Amazon Simple Queue Service (SQS) などのサービスからの通知を利用し、ファイルをリッスンして、到着次第ロードします。Snowpipeは、ファイルがステージに追加されてから数分以内にデータをロードして取り込めるように転送するため、ユーザーは、生データが入手可能になり次第直ちに最新の結果を得ることができます。

AWS、Azure、Googleのオブジェクトストアに新しいデータが届くたびに継続的にデータがロードされるため、スクリプトやスケジューリングツールは不要です。Snowpipeを活用したデータパイプライン運用により、ストリーム内の自動タスクやCDC情報を使いながら継続的にデータのマイクロバッチをステージングテーブルにロードして変換・最適化できます。

Snowpipeによるデータ取り込みのメリットを図5で説明します。



## Kafka用Snowflakeコネクタを使用したイベントストリームロード

真に価値のあるデータを得るには、まずデータを収集し、保存し、分析する必要があります。Apache Kafkaは、イベントデータの確実な収集、送信、提供のために多くのユーザーに選ばれているシステムです。

Kafka用Snowflakeコネクタを利用すると、Kafka Connectクラスターを簡単に構成でき、JSONやAvroのイベントをSnowflakeテーブルに取り込むことができま

す。このコネクタは、1つまたは複数のApache KafkaトピックからレコードをSnowflakeの内部ステージに継続的にロードし、Snowpipeを使用してステージングテーブルにロードします（図6参照）。イベントがSnowflakeにロードしたら、充実したデータパイプライン機能を駆使してデータをさらに処理したり、他のビジネスデータと統合したり、アナリティクスに使用できるようにデータを改良したりすることができます。

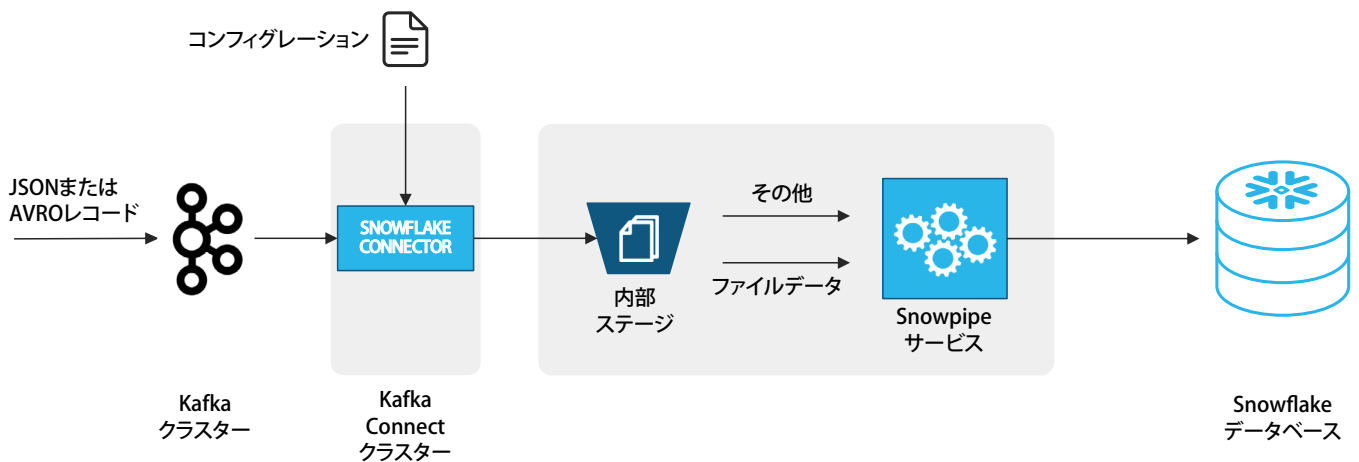


図6: Kafka用Snowflakeコネクタは、JSONやAvroのイベントをSnowflakeテーブルに自動的に取り込みます

## Snowflakeデータマーケットプレイスからのライブデータローディング

Snowflakeデータマーケットプレイスは、Snowflakeの安全なデータシェアリングを利用してデータプロバイダーと利用者をつなぎます。データサイエンス、アナリティクス、エンジニアリングの各チームは、さまざまなサードパーティデータを検出してアクセスし、それらのデータセットをSnowflakeアカウントで直接的に利用し、変換なしで照会したり、自社のデータと結合したりすることができます。

Snowflakeデータマーケットプレイスを利用すれば誰でも、クエリ対応のライブデータにアクセスしてデータドリブン型意思決定を行うことができます。マーケットプレイス上で利用できるデータは、お客様のビジネスパートナーや顧客で構成されるエコシステムからのデータに加えて、最大数千社の外部データプロバイダーやデータサービスプロバイダーからのデータも含まれます。FTPやAPIは一切不要です。

Snowflakeデータマーケットプレイスへの参加により以下のメリットが得られます。

- 管理・共有されたライブデータセットに安全にアクセスし、ほぼリアルタイムで自動更新を受信できる
- 古いデータのコピーや移動が必要なために生じるリスクや煩雑さを一掃する
- 新しいデータセットをSnowflake内の既存のデータと組み合わせ、新たなビジネスインサイトを取得できる
- ユーザーにとっては、データセットがすぐに利用でき、継続的に更新される
- ベンダーからの生データ製品にスムーズにアクセスできる
- データのロードや更新に使用する、さまざまなAPIとデータパイプラインを構築・維持するコストを排除する
- 自分の好きなBIツールを使用する
- サードパーティのデータソースを見つけてテストする

## SNOWFLAKE内のデータを変換してデータのパワーを引き出す

SnowflakeはETLとELTの両方のデータロードに対応していますが、より多くのデータ変換をSnowflakeに取り入れることで以下のような豊富なメリットが得られます。

- Snowflakeの強力な処理エンジンによる、スムーズで信頼性の高いデータ処理
- 瞬時のスケーラビリティとリソース競合の排除によりデータワークロードに対応
- 半構造化データをネイティブサポート
- データキャプチャ機能の自動切換え
- ストアドプロシージャ
- 外部関数
- Java UDFにより、そのままSnowflake内でJavaコードを実行
- Snowpark APIによるデータプログラミング
- 非構造化データに対応

### 強力な処理エンジンにより、データを今ある場所から動かさず、そのまま変換、加工、拡充ができます。

Snowflakeのデータクラウドは、処理対象データとリジッドに連結はされない(デカップルド)一方でデータとの密接な統合を実現する強力な処理エンジンを構築します。

これによりほぼ無制限のデータ量をお好きな言語で確実かつスピーディーに処理できるようになります。図7に示すとおり、このユニークな特長を備えた処理エンジンは以下のメリットを提供します。

- **重複したデータ処理を排除。**一度処理したデータはすべての分析ユースケースで利用可能になり、処理したデータセットに社内からも社外からもスムーズにアクセスできます。いつでも紹介できるように整理されたライブデータに瞬時にアクセスでき、自由にデータを拡充できるため、生産性が向上します。
- **複雑性を最小限に抑えた思いのままのシステム構築。**ScalaやJava(プレビューモード)などほとんどの一般的なプログラミング言語をサポートするフレキシブルなデータパイプラインを構築します。お好きなIDEでコードを作成し、プッシュダウン機能によりSnowflake内でデータ処理。複雑性を抑えたアーキテクチャにより、非SQLパイプラインの運用のための追加環境の管理が不要になります。
- **優れたスピードと信頼性の稼働性能。**自動マイクロパーティショニングやデータクラスタリングにより高効率でのデータ処理を実現。マルチクラスター構造のワークロードもサポートする自動スケールポリシーによりどのような規模のユーザー数やジョブ数にも瞬時に対応。ワークロードの自動スケール・サスペンド・再開により最も優れたコスト効率で処理ニーズに対応します。



### 重複したデータ処理を排除

データクラウド上の整理されたライブデータを共有・消費、必要に応じて補完・強化



### 複雑性を最小限に抑えた思いのままのシステム構築

あらゆるデータを処理でき堅牢なトランスフォーメーション能力を備え、使用言語の選択肢が豊富



### 優れたスピードと信頼性の稼働性能

信頼性・スケーラビリティ・コストパフォーマンスを組み込んだ強力な処理エンジンによるパフォーマンスの自動化

図7: Snowflakeの処理エンジンは、信頼性の高いデータ処理をスムーズに実行します

## マルチクラスターのコンピューティング アーキテクチャが伸縮性と拡張性を提供

Snowflakeのデータパイプラインはバッチまたは継続のいずれかを選択でき、Snowflake上で直接データを処理できます。Snowflakeではマルチクラスターのコンピューティングアプローチを採用しているため、これらのパイプラインはその他のワークロードの性能に影響を与えることなく複雑な変換に対応できます。

その仕組みについて説明します。Snowflakeのマルチクラスターでは、共有データアーキテクチャ、コンピューティングリソースやストレージリソースは物理的には別物ですが、論理的にはこれらすべてが一元化された1つの統合データウェアハウスシステムを構成しています。この独自のマルチクラスターアーキテクチャにより、ユーザーニーズに応じた必要数の個別ワークロードに対応できます。各ワークロードにそれぞれ専用のコンピューティングエンジンが割り当てられ、必要に応じて規模の拡大/縮小ができます。予めリソースを割り当てたり、他のプロセスを中断したりする必要はありません。

この素晴らしいアーキテクチャにより、Snowflakeは複数の多種多様なワークロードに簡単に対応できます。コンピューティングとストレージの分離により、多様なサイズの仮想ウェアハウスをスピンアップして、ELTプロセスを実行でき、BIレポートユーザー、データサイエンティスト、データマイナーをサポートできます。リソースの競合はまったく発生しません。

## 半構造化データをネイティブにサポートし 変換アーキテクチャを簡略化

JSON、XML、Avroなどの半構造化データ形式は、必ずしも明確に分類されているものではありません。通常は、ノードのヒエラルキーがあります。それぞれのノードには名前と値があります。これは、単一の定数、名前/値のペアの列、あるいはネストされたオブジェクトのいずれかとなります。従来のデータベースでは、異なる構造を扱うことができないため、開発者はETLツールに頼り、データを従来の行/列の構造にフラット化する複雑な変換を設計することになります。

Snowflakeでは、VARIANTという半構造化データに特化したデータ区分を設定しています。ここで、半構造化データの取り込みプロセスの仕組みをご紹介します。仮想ウェアハウスは、JSONまたはXMLのドキュメント全体をそのままの状態にVARIANT列に書き込みます。ユーザーがSQLでその列をクエリすると、同じJSONまたはXMLのドキュメントが戻ります。

同時にグローバルサービス層がドキュメントのコンテンツに関する特定のメタデータ(ノード名、ヒエラルキー、配列構造)を収集し、そのデータをメタデータストアに

書き込みます。Snowflakeは、ユーザーがSQL内で直接データを参照できるよう、内部でデータを列に整理します。この列データはマイクロパーティションファイルに書き込まれますが、テーブル定義には実際の列名は表示されません(JSONノードの列名は存在しないということになります)。VARIANT列のノードを参照するには、SQLの単純なドット記法を使用します。このプロセスは性能が高いため、変換内容を完全に記述する必要がありません。

## CDC機能が変更内容を追跡

Snowpipeで継続的なデータパイプラインを構築し、Snowflakeストリームアンドタスクを使用してデータ統合ジョブをスケジューリングし、変更データを取得できます。

- ストリームはSnowflakeオブジェクトタイプ的一种であり、挿入やDMLの変更など、テーブルの変更の差分を追跡するCDC機能を提供します。これにより、変更されたデータを使ってアクションを起こすことができます。テーブルストリームでは、2つのトランザクション間でテーブルを照会し、テーブルに対し行単位で一連の変更を反映させることができます。
- タスクもSnowflakeのオブジェクトタイプで、SQLステートメントを実行するための定期的なスケジュールを定義します。これには、ストアプロシージャを呼び出すステートメントが含まれます。タスク同士をつなげて連続実行すると、より複雑な反復処理に対応できます。

継続的なデータパイプラインでは、タスクはオプションでストリームを使用して新規または変更されたデータを継続的に処理する便利な方法を提供する場合があります。タスクは、ストリームにテーブルの変更データが含まれているかどうかを検証し、変更データがある場合にはこれを利用し、変更データが存在しない場合は現在の実行をスキップできます。

変換終了後、変換されたデータを、データレイクエクスポート機能を使用してデータレイクにアップロードし直すことができます。Snowflakeストリームアンドタスクを利用すると、データ統合ジョブのスケジューリングと変更データの取得を簡単に実行できます。そのため、毎回すべてのデータをロードする必要がなくなり、変更データのみを処理するだけでOKとなります。

## ストアプロシージャによる頻繁なタスクの自動化

プロシージャにより、頻繁に実行されるタスクで、複数のSQLステートメントを必要とするものが自動化されます。プロシージャは、一度作成しておけば何度でも実行できます。

ストアプロシージャを作成するにはJavaScriptと、多くの場合はSQLを利用します。JavaScriptが制御構造(分岐とループ)を提供し、JavaScript APIの機能呼び出すことでSQLが実行されます。

ストアプロシージャで提供される内容は次のとおりです。

- ストレートSQLではサポートできない手続き型ロジック(分岐やループ)
- エラー処理
- SQLステートメントを動的に作成し実行
- プロシージャ実行者役割の特権ではなくプロシージャ所有者役割の特権によるコード作成

## Snowflake外部関数によるデータパイプラインの拡張

データ変換の多くは非常に複雑で構築が困難です。たとえば他の言語やフレームワークを使って構築する場合もあれば、サードパーティのコードや外部サービスを活用する場合があります。以前は、複雑なアーキテクチャを持つさまざまなデータ環境で、さまざまなサービスやシステムを管理する必要がありましたが、Snowflakeではパ

イプラインを拡張できるため、外部関数の定義やサードパーティサービスの活用が可能となります。

ユーザーは自身のリモートサービスの書き込みや呼び出し、サードパーティ製のリモートサービスの呼び出しができます。これらのリモートサービスは、AWS Lambdaなどのクラウドサーバレスコンピューティングサービスを含め、あらゆるHTTPサーバスタックを使って書き込むことができます。

外部関数の便利な使用例を以下に紹介します。

- ジオコーディングサービスを利用して住所を座標や政治的地域で増補する
- サードパーティサービスを利用してメッセージのセンチメント分析を実行する
- カスタム機械学習モデルを利用して顧客をスコアリングする
- カスタムロジックを利用してメールからメールアドレスを抽出する
- リモートサービスからライブの株価を取得する

外部関数の利用法とメリットを次のテーブルにまとめました。

外部関数： お使いのデータパイプラインでカスタムまたはサードパーティのサービスや変換ロジックを利用し、 データの変換や増補を実行します	
利用ケース	<ul style="list-style-type: none"><li>• パイプラインで複雑なデータ変換の能力に限りがある場合</li><li>• これらの活動をSnowflakeの外部で実行する要件がある場合</li></ul>
メリット	<ul style="list-style-type: none"><li>• <b>開発者</b>はSnowflake内の関数としてウェブサービスをエクスポートできます</li><li>• <b>ユーザー</b>はSnowflake内に組み込まれているものと同様にエクスポートされた機能にアクセスできます</li><li>• 一方で<b>管理者</b>は自社システム内のデータの行き先を管理できます</li></ul>

## Java UDF(ユーザー定義関数)によるSnowflake内でのJVM実行

Java UDF(ユーザー定義関数)の使用により、Javaメソッドを記述してSQL関数と同様に呼び出すことができます。これによりカスタムロジックを使用してデータを変換し増強しながら併行してジョブを実行できるため、別のサービスを管理することなくパフォーマンスを向上できます。

Java UDFのメリット:

- 開発者は一般的なJava言語とライブラリを使用してSnowflake内に機能を組み込むことができます
- ユーザーはSnowflakeに元々組み込まれている場合と同様にこれらの機能にアクセスできます
- 一方管理者は、データがSnowflakeの内部に留まり外に出ないという安心感をいただけます

## Snowpark APIによるデータプログラミング性能の拡張

Snowpark(現在プレビューステータス)は、好みの言語でSnowflakeのコードを作成し、Snowflake内で直接実行できる新しい開発者向けの機能です。図8のように、Snowparkの導入により以下のメリットが得られます。

- 自分のプロジェクトや好みに合わせた使い慣れた言語でコードを作成。SnowparkにScalaサポートが追加されました。今後Javaその他の言語についてもサポート追加予定です。
- データパイプラインの作成とデバッグをスムーズに完了できます。またカスタムライブラリやサードパーティライブラリを導入すれば新機能の作成や既存機能の登録も可能です。
- 追加システムの設定やクラスターのスピニングを行わずにSnowflake内でワークロードを実行。

Snowpark APIは、Snowflake処理エンジンの魅力である「ニアゼロメンテナンスでの優れたパフォーマンス、信頼性、スケーラビリティの提供」を最大限に活用するべく設計された機能です。Snowpark APIの利用により、使い慣れた言語と密接にリンクされたコード作成が

Snowflake上で可能となります。これによりSnowflakeの真のデータプログラミング性能を開発者に開放するとともに、すべてのユーザーがSnowflakeの強力な処理エンジンの利便性を享受できるようになります。対応言語のユーザーは、それぞれ自らの開発ノウハウと密接にリンクされたコード作成ができます。不透明で分かり辛いSQLではなく自分の使い慣れた言語でコードを作成することで、好きなIDEのメリットを最大限に活かしながらインテリセンス、コード補完、タイプチェックを利用できます。使い慣れた言語を使用することで、SQLよりはるかにスムーズに多様なシナリオを表現でき、デバッグも容易になるでしょう。

Snowpark APIはSnowflakeのエンジン内にオペレーションをプッシュダウンすることで動作するため、他のシステムのビルドアウトや管理は不要です。Snowflakeでは今後Javaサポートの追加を予定していますが、このプッシュダウン機能を利用すればJavaサポートのメリットを最大限に活かしたカスタムリソースのプッシュダウンが可能になり、たとえばサードパーティ製ライブラリやカスタムライブラリなどをSnowflake内へのプッシュダウンできます。プッシュダウン機能によりデータをSnowflake外部に移動させることなくデータ変換が可能になるため、開発者はSnowflakeがネイティブサポートしていない機能もシームレスにSnowflakeに統合できます。

## 非構造化データのサポートによりプラットフォーム性能を強化

音声、動画、PDF、画像データといった非構造化データが急増しています。Snowflakeは、現在プレビュー段階ではありますが非構造化データをサポートしています。今後非構造化データのサポートが一般機能として確立されれば、データエンジニアはあらゆる非構造化データのパイプライン実行をオーケストレーションできるようになり



## デベロッパー • データエンジニア • データサイエンティスト

自分のやりやすい方法でSnowflakeのコードを作成し、Snowflakeで直接実行できる新しいデベロッパー向け機能

### 効率に優れた強力なパイプライン

使い慣れたコンストラクトで簡単にデータパイプラインの作成とデバッグを行い、サードパーティライブラリを統合

### 言語の選択

好みの言語とツールを用いてコードを作成

### 1つのシステムで管理

他の処理システムでの実装は不要、Snowflake上で直接運用

図8: Snowparkのデータプログラミング性能により、データエンジニアはデータパイプラインの機能と性能を大幅に拡充できます

ます。Snowflake内での非構造化データの管理により複数の異なるシステムへのアクセスや管理が不要になり、各種ファイルや非構造化データ、付随するメタデータに対するきめ細かなガバナンスを展開できるようになります。またより広範なビジネスユースケースの実現により新たな収益チャンスの可能性も生まれるでしょう。

### ユーザーへのデータ提供

データの変換後、データエンジニアは変換されたデータが組織内外のユーザーに適切に提供されるかどうか確認する必要があります。

### データクラウドでの共有とコラボレーション

企業は、社内では他のビジネスユニットや子会社と、社外ではベンダー、パートナー、サプライヤー、顧客とデータを共有する必要があります。共有したデータにより、イノベーションが促進され、データに基づく意思決定がより良いものとなり、トレンドを予測するための新たなインサイトが得られます。しかしリスク、セキュリティ、プライバシーなど、データコラボレーションに付随する問題が、効果的なデータシェアリングの障害になる場合もあります。

従来のデータシェアリングの方法（API、ETL、FTPなど）はデータの移動やコピーを伴うためコスト増とセキュリティ問題につながるだけでなく、得られるデータはタイミングが古く同期されないものでした。

一方、Snowflake独自のマルチクラスター型共有データアーキテクチャが実現するSnowflake Secure Data Sharingは、データのコピーや移動を伴わない安全なデータシェアリングを可能にします。Secure Data Sharingのユニークな特長は、複数のクラウドや地域にまたがるグローバルなデータシェアリングをサポートする点です。データ、ロジック、サービスの共有を可能にし、必要に応じたアクセスの取り消しを含めた安全なデータシェアリングを実現できます。

次ページの図9に示すとおり、Secure Data Sharingには以下のメリットがあります。

- 150社以上のプロバイダーが提供するデータやサービスにアクセスでき、Snowflakeデータマーケットプレイスを介して製品を販売・提供できる
- SnowflakeのDirect ShareとData Exchangeの機能、リーダーアカウントとの組み合わせにより、お客様のビジネスエコシステム全体でのデータシェアリングができる

SnowflakeのSecure Data Sharingは、Snowflakeデータマーケットプレイスの技術基盤です。Snowflakeデータマーケットプレイスは、データサイエンティスト、

「……組織的な境界や地理的境界をまたいでデータをつなぎ合わせれば、年間約3兆ドルの経済価値が見込めます……」

### McKinsey & Company

共通の利益のためのコラボレーション  
Navigating public-private data partnerships (公益のためのコラボレーション：官民のデータパートナーシップをナビゲートする)、2019年5月30日<sup>2</sup>

## データクラウドでの共有とコラボレーション - あらゆるシナリオにおける安全なデータシェアリング

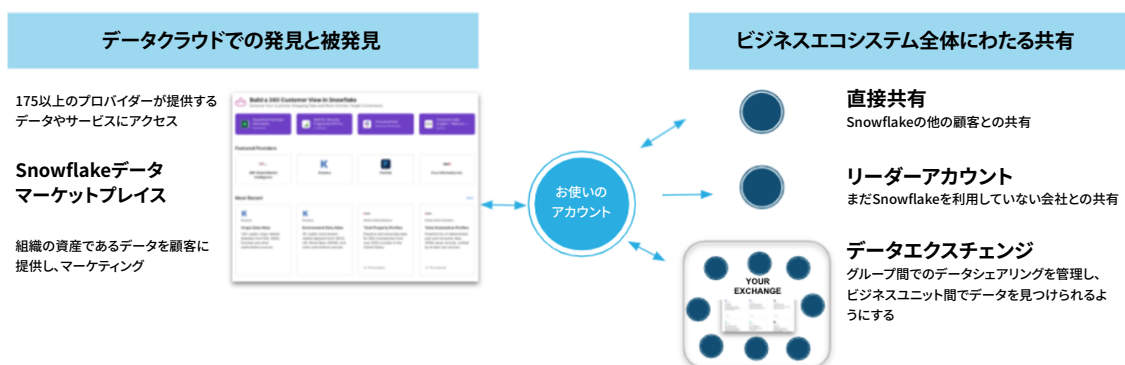


図9: Secure Data Sharingは瞬時のコラボレーションを安全に実現

ビジネスインテリジェンスやアナリティクスの専門家、データドリブン意思決定を求めるすべての人向けに、ビジネスパートナーや顧客で形成される事業エコシステムだけでなく、最大数千社ものデータプロバイダーやデータサービスプロバイダーから得られるクエリ対応のライブデータへのアクセスを提供します。

Snowflakeデータマーケットプレイスを利用する企業は、プライベートデータハブを構築し、従業員、ビジネスユニット、パートナー、顧客などにアクセス権を付与することができます。Snowflakeのプラットフォームを利用すると、データの移動、コピー、転送はほぼ不要となります。さらに企業は、Snowflakeデータマーケットプレイスのパブリックデータセットを活用し、自社データと組み合わせた分析や自社のデータハブ内での共有もできます。

Snowflakeデータマーケットプレイスのメリット：

- 迅速かつ簡単なデータソーシング - 古いデータのコピーや移動に伴うリスクや煩雑さを一掃します。管理・共有されたライブデータセットにセキュアにアクセスし、ほぼリアルタイムで自動更新を取得できます。
- アナリティクスコストの削減 - Snowflakeアカウントから直接、安全かつ管理された環境でクエリに対応した共有ライブデータにアクセスできるため、データの取り込みや変換といった従来のETLプロセスに付随するコストや労力がほぼ完全に排除されます。
- 独自データの収益化 - データプロバイダーとしてSnowflakeデータマーケットプレイスに参加することにより最大数千社ものSnowflakeデータ利用者に自社管理データアセットを販売できるため、新たな収益源を創出できます。

Snowflakeデータマーケットプレイスの詳細については、[こちら](#)をご覧ください。サードパーティータを活用したアナリティクスに関するベストプラクティスについては、こちらの[eブック](#)をご覧ください。

### Snowflakeのロバストなパートナーエコシステムの活用

Snowflakeは高度に統合された機能を備えた充実したパートナーエコシステムを提供し、お客様が抱えるさまざまなデータニーズに対応します。このような認証済みパートナーとの連携や統合は、お客様がSnowflakeのフレキシビリティ、パフォーマンス、利便性を活かしてより有意義なデータインサイトを提供するのに役立ちます。Snowflakeパートナーネットワークには、テクノロジー、サービス、クラウドストレージ、サードパーティータを提供するパートナー企業が多数、参加しています。

まずはSnowflakeパートナーコネクトを利用してネットワークに参加してみましょう。パートナーコネクトを利用すると、[指定した](#) Snowflakeビジネスパートナーとのトリアルアカウントを作成し、Snowflakeに統合することができます。この機能は、さまざまな追加ツールやサービスを試した上でお客様のニーズに最適な選択肢を見つけない場合に非常に便利です。

またパートナーコネクトを利用することで、より迅速にデータロードを開始できます。Snowflakeのテクノロジーパートナーとの統合が事前に構築されているため、オンボーディングプロセスが簡略化されます。パートナーのアプリケーションのプロビジョニングや構成を自動的に実行し、数分でSnowflakeへのデータロードを開始しすぐに分析できます。

## データレイクへのデータのエクスポート

アーキテクチャの設計方法によっては、外部テーブルを使用してクラウドデータレイクに直接アクセスできます。あるいは、Snowflakeのサーバレス取り込みシオンサービスであるSnowpipeを利用して、データを自動的にSnowflakeに取り込むこともできます。Snowflakeストリームアンドタスクを利用すると、データ統合ジョブのスケジューリングと変更データの取得をスムーズに実行できます。毎回すべてのデータをロードする必要がなくなり、変更データのみを処理するだけでOKです。変換が完了したら、Snowflakeのデータレイクエクスポート機能を使い、選択した列で自動的にパーティショニングされたデータレイクに、データをアンロードできます。

エクスポート機能により、たとえばSnowflakeで大量の生データを処理し、処理したデータを簡単にデータレイク内にエクスポートし直すことができます (Parquet、CSV、JSONに対応)。

## 一般的なドライバへの接続

Snowflakeは、Python、Spark、JDBC、ODBCなどの一般的なドライバを提供するよう設計されており、下流のアプリケーションへのデータ持ち込みが容易でさまざまな分析ニーズに対応できます。

- Python用Snowflakeコネクタは、Snowflakeに接続しすべての標準オペレーションを実行できるPythonアプリケーション開発用インターフェイスを提供します。Python用Snowflakeコネクタは、JDBCやODBCドライバを使ったJavaやC/C++でのアプリケーション開発に代わる代替プログラミング手段となります。
- Spark用Snowflakeコネクタの導入によりApache Sparkエコシステム内にSnowflakeが組み込まれ、SparkからSnowflakeのデータを読み書きできるようになります。Sparkから見ると、Snowflakeは他のSparkデータソース (PostgreSQL、HDFS、S3など) と同じように見えます。Sparkコネクタは、SnowflakeクラスターとSparkクラスター間の双方向データ移動をサポートします。Sparkコネクタを使用すると、読み込みや書き込みなどの操作を実行できます。たとえばSpark DataFrameに入力したり、Spark DataFrameをSnowflakeテーブルに書き込んだりすることができます。
- Snowflake JDBCドライバは、コアJDBC機能に対応しています。Snowflake JDBCドライバは、データベースサーバへの接続にJDBCをサポートしているほとんどのクライアントツール、アプリケーションと併用できます。JDBCを利用すると、データソースとの接続を確立し、クエリや更新ステートメントを送信し、結果を処理することができます。
- Snowflake ODBCドライバを利用すると、ODBC接続をサポートするすべてのアプリケーションから直接Snowflakeにアクセスできます。ODBCドライバは、SQLクライアントなどの各種クライアントと併用可能です。



# SNOWFLAKEについて

Snowflake は、Snowflake のデータクラウドを用い、あらゆる組織が自らのデータを活用できるようにします。お客様には、データクラウドを利用してサイロ化されたデータを統合し、データを発見してセキュアに共有し、多様な分析ワークロードを実行していただけます。データやユーザーがどこに存在するかに関係なく、Snowflake は複数のクラウドと地域にまたがり単一のデータ体験を提供します。多くの業界から何千ものお客様（2021年7月31日時点で、2021年 Fortune 500 社のうちの212社を含む）が、Snowflake データクラウドを自社のビジネスの向上のために活用しています。詳しくは [www.snowflake.com](http://www.snowflake.com) をご覧ください。



© 2021 Snowflake Inc. All rights reserved. Snowflake、Snowflakeのロゴ、および本書に記載されているその他すべてのSnowflakeの製品、機能、サービス名は、米国およびその他の国におけるSnowflake Inc.の登録商標または商標です。本書で言及または使用されているその他すべてのブランド名またはロゴは、識別目的でのみ使用されており、各所有者の商標である可能性があります。Snowflakeが、必ずしもかかる商標所有者と関係を持ち、または出資や支援を受けているわけではありません。