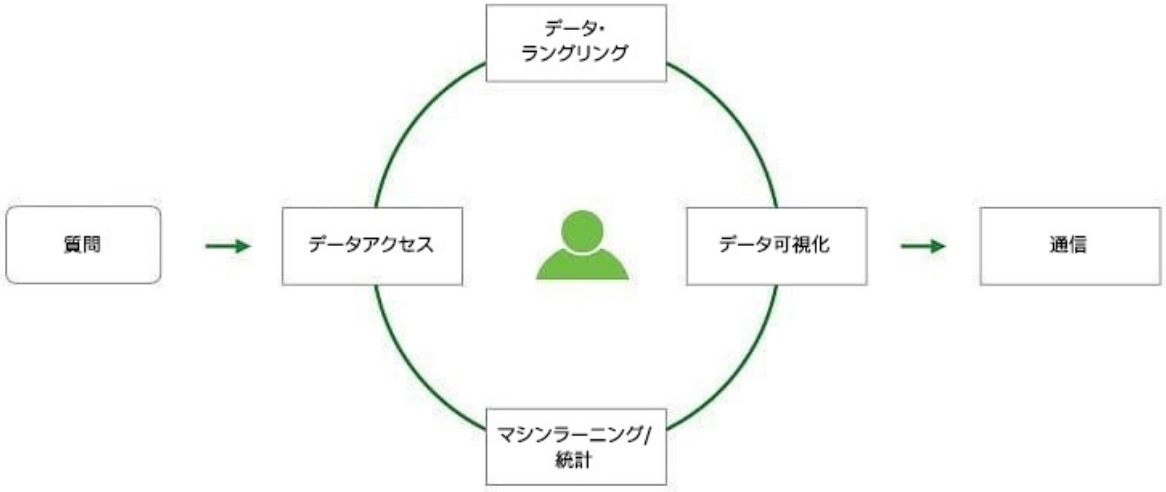
データ・サイエンス・アプリケーションを次のレベルへ

スケーラビリティー、パフォーマンス、柔軟性

Venkat Krishnamurthy OmniSci プロダクト・マネージメント・バイスプレジデント

この 10 年間、テクノロジーの展望は主にデータ自身により推進されてきたことは明らかです (**データと AI の状況 2019** (英語) を参照)。純粋な応用科学であれ産業であれ、人間の試みのあらゆる分野で、問題を解決する主な方法として、我々はデータを収集して使用しています。その結果、データサイエンスが重要な分野として注目を集めるようになりました。増え続けるデータセットから有用な情報を整理して引き出せることは重要なスキルセットであり、データ・サイエンティストがより大きくより詳しいデータを扱えるようにさまざまなツールと手法が登場しました。

データ・サイエンティストの一般的なワークフローは、基本的に**図 1** のような反復プロセスです。

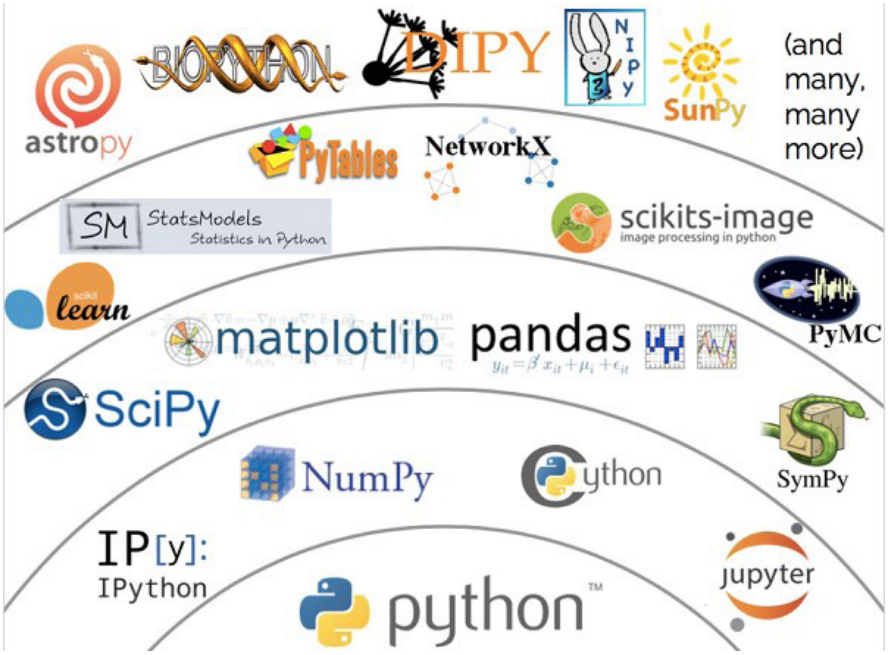


1 **データ・サイエンス・ワークフロー**

データ・サイエンティストは、(職業プログラマーではなくデータ・サイエンティストにとっての) 使いやすさと統計および数値計算向けライブラリーを広範にサポートするエコシステムの組み合わせとして、Python* および R エコシステムを長年支持してきました。近年、データサイエンスの重要なサブフィールドとしてディープラーニングと AI が出現したことで、これらのエコシステムにも多くの機能が追加されています。特に Python* は、マシンラーニングと AI ワークフローで広く普及しています。

Python* の世界では、しばらくの間、PyData スタック (**図 2**) が最も完全で人気の高いデータ・サイエンス・ツールのセットでした。N 次元配列データの最も低いレベルの数値計算 (NumPy) から始まり、このスタックは、科学計算 (SciPy*)、表形式 / リレーショナル・データ解析 (pandas)、シンボリック計算 (SymPy) の一連のレイヤーを提供します。

スタックの上位には、可視化 (Matplotlib、Altair*、Bokeh)、マシンラーニング (scikit-learn)、グラフ・アナリティクス (NetworkX など) に特化したライブラリーも用意されています。AstroPy や BioPython などのドメイン固有のツールキットは、これらのレイヤー上に構築され、オープンツールのディープでリッチなエコシステムを提供します。これらに加えて、Jupyter* プロジェクトは、インタラクティブ・コンピューティングのアイデア全般、特にデータに基づくストーリーテリングの推進に大きく貢献しました。多くのデータ・サイエンティストは、デフォルトの開発環境として Jupyter* を使用し、仮説とモデルを作成して調査しています。



2 **Python* データサイエンス (PyData) スタック**

OmniSci: 最新のハードウェアを活用してアナリティクスを高速化

OmniSci (旧 MapD) では、2013 年以降、HPC の手法を使用して分析 SQL とデータ可視化を同時に高速化しました。オープンソースの OmniSciDB SQL エンジンは、複数のアイデアを結集したものです。

- メモリー階層の効率的な利用による I/O アクセラレーション
- 分析 SQL カーネル向けの LLVM ベースの JIT コンパイル
- 大規模なインサイチュ・データの可視化
- マシンラーニングやディープラーニングなどのアウトオブコアのワークフローとの効率的なデータ交換

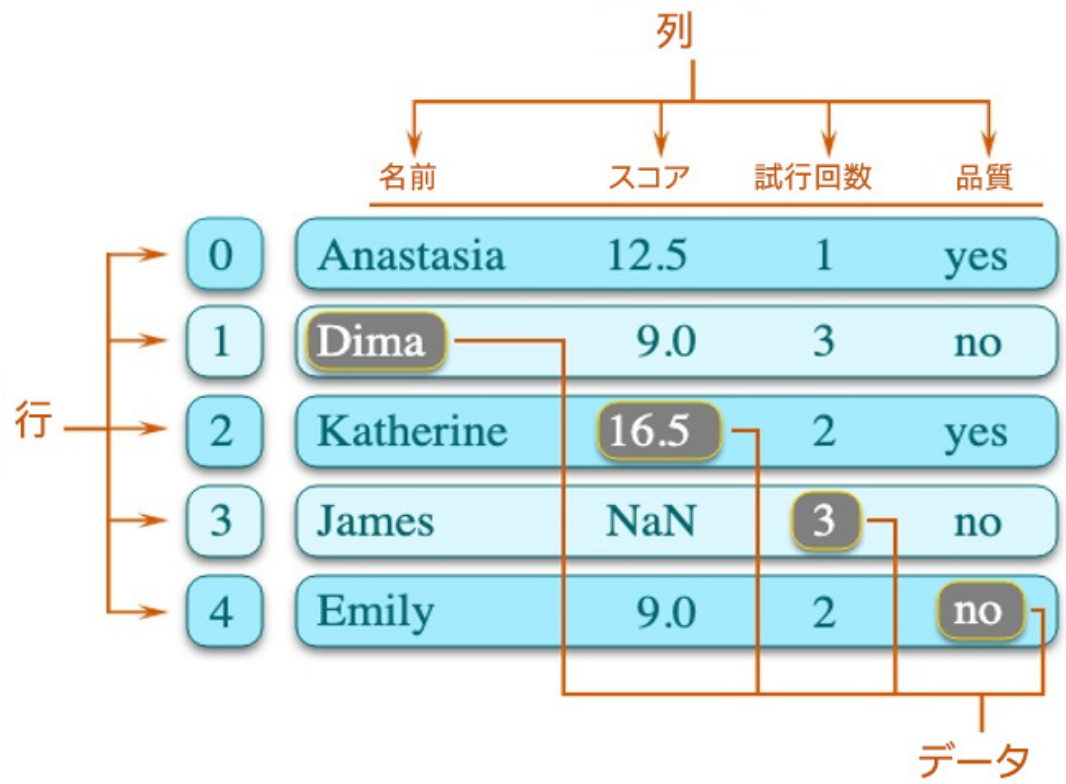
これらのアイデアを結集することにより、GPU などのハードウェア・アクセラレーターと最新のベクトル CPU の両方で、分析 SQL クエリーを 2 桁から 3 桁高速化できます。

OmniSci は当初、分析 SQL とデータ可視化の問題に取り組んでいましたが (OmniSciDB エンジンは 2017 年にオープンソース化)、主要な統合とインターフェイスを備えたオープンな PyData エコシステムにさらなる価値を提供できることに気付きました。我々は、データ・サイエンティストが PyData スタック内のプログラム・ワークフロー、つまり Jupyter* の内部で作業して、(未加工の SQL ではなく) 使い慣れた API を使用してデータを操作できるようにしようと考えました。これを念頭に、NumPy と SciPy* の作者の Travis Oliphant 氏により設立された Quansight Labs と協力して、Python* API のレイヤーのセットでコア OmniSciDB エンジンのスケーラビリティーとパフォーマンスを活用するオープン・データ・サイエンス・スタックを実現しました。

データフレームと関連する問題

その際に、分析データ構造に関連する、基本的なデータ・サイエンス・ワークフローでいくつかの問題に遭遇しました。テンソル (マシンラーニングやディープラーニングの主要なデータ構造であるデータの多次元配列) に相当するデータフレーム (**図 3**) は、テーブルの既存のリレーショナル・データベース・パラダイムに密接にマップされる、おそらくアナリティクスで最も一般的に使用されるデータ構造です。

データ・サイエンティストはあらゆる種類の表形式データからデータフレームを作成します。データフレーム・ライブラリーは Python* にも存在し、最も人気の高いのは pandas です。R (これらの起源) や Julia などの新しい言語でも同様です。



3 **データフレームの構造**

問題は、エコシステム全体のアナリティクス・ツールの数が急増するとともに、データフレームの実装数も増加することです。例えば、Apache Spark* は非常に人気の高い解析処理エンジンおよびプラットフォームです。多くのデータフレームの機能を提供するデータフレーム API (Spark* の分散データセット) を備えていますが、pandas とも、R とも異なります。OmniSci では、これらの API はすべて非常に柔軟であるにもかかわらず、データ・サイエンティストが非常に大規模なデータセットのインタラクティブな調査を行うことを困難にする、スケーラビリティーとパフォーマンスの問題に悩まされていることが分かりました。

例えば、Spark* では非常に大きな分散データフレームが可能ですが、Spark* は何よりもまず分散システム向けに設計されている (そして Java* 仮想マシンでの実行には追加のオーバーヘッドが発生する) ため、計算エンジンは真にインタラクティブな (1 秒未満の) クエリーを処理できません。一方、pandas はリッチで強力な API ですが、操作が Python* で実装されており、インタープリター環境での実行により大幅なオーバーヘッドが発生するため、スケーラビリティーの問題があります。

最後に、言語やエコシステム (JVM、Python*、R) にわたる既知の交換の問題があるため、さまざまな計算環境に分散できる完全なワークフローを構築することは困難で非効率的です。例えば、Spark* は、Python*/pandas などのローカル・コンピューティング環境で、詳細な解析のために大規模なデータセットを操作および形成するため、ワークフローの初期段階でよく使用されます。最近まで、これによるスケーラビリティーを制限する交換と変換のオーバーヘッドが発生していました。幸いにも、このニーズに対応するデファクト・スタンダードとして Apache Arrow* が登場しました。しかし、異なるフレームワーク間のデータ交換での採用はまだ進んでいません。

データサイエンスへの取り組みを始めて間もなくしてから、OmniSci は Ibis に参加しました (**図 4**)。pandas の作者の Wes McKinney 氏が開発に携わっている API である Ibis は、OmniSci などの大規模なデータ処理およびストレージシステムと Python* データ・サイエンス・スタックをつなぐ興味深い方法を提供します。この方法では、PyData スタックとデータストアの世界をつなぐ、いくつかの主要なレイヤーが提供されます。Ibis プロジェクトのウェブサイトによれば、次のコンポーネントが含まれます。

- 構造化データに対して構成可能で再利用可能なアナリティクスを可能にする、アナリティクス向けに特別に設計された **pandas などのドメイン固有の言語 (DSL)** (Ibis 表記)。SQL SELECT クエリーで表現できることは Ibis で記述できます。
- HDFS およびその他のストレージシステムへの**統合ユーザー・インターフェイス**。
- 複数の SQL システムをターゲットとする**拡張可能なトランスレーター・コンパイラー・システム**。

4 **データサイエンスとデータストアをつなぐ API の Ibis**

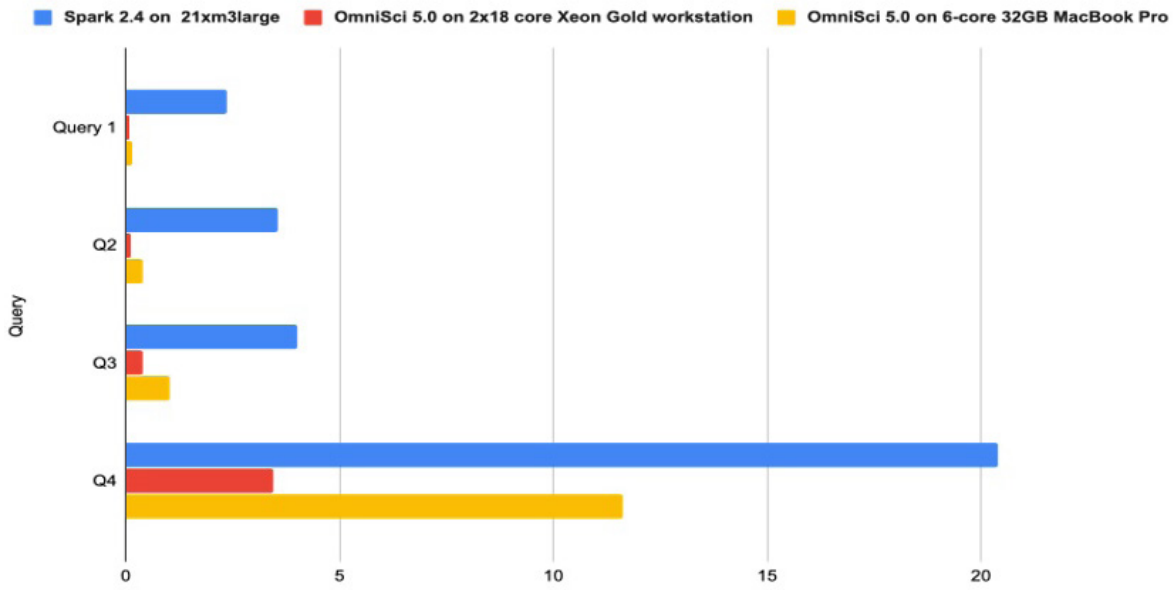
MySQL*、Postgres*、BigQuery*、OmniSci、Spark*、Clickhouse などを含む SQL システムをサポートし、新しいバックエンドを簡単に追加できます。

また、宣言型の Python* 可視化フレームワーク、Altair* で構築されるオープンデータの可視化にも投資しました。Altair* は、データの可視化に最新の宣言型のユーザー・インターフェイス・フレームワーク、Vega および Vega-Lite を利用します。Quansight と協力して Ibis と Altair* を統合することにより、ソースシステムからデータを移動することなく、非常に大きなデータセットを可視化して調査できるようになりました。

OmniSci とインテル : データサイエンスにおける連携を強化

2019 年に、これらの統合の最初のバージョンをリリースしました。これらもすべて、各プロジェクト内でオープンソース化されています。その際に、オープンソースの OmniSciDB エンジンの能力とパフォーマンスを目にしたインテルと、非常に有益な共同作業を開始することになりました。我々はともに、データフレーム中心の分析ワークフローを高速化するため、共通の目標として、その機能とデータ・サイエンス・ツールの橋渡しとなる方法を検討しています。インテルのチームは、エコシステムの能力を示すリファレンス・プラットフォームとして OmniSci を選択しました。詳細なシステムとチューニングの専門知識を備えたインテルのチームは、この目標に向けて大きく貢献しています。

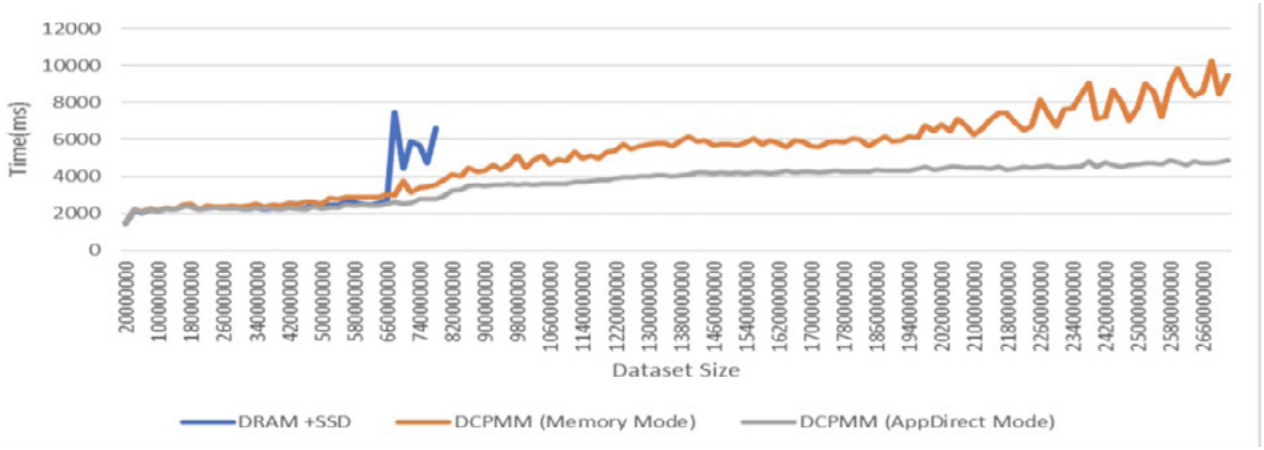
チームは、**インテル® VTune™ プロファイラー**などのインテルのプロファイリング・ツールを使用して、インテル® CPU ファミリーで OmniSciDB 実行 (**インテル® スレッディング・ビルディング・ブロック**とロックの使用を含む) のパフォーマンス・チューニングと最適化を行うべき領域を特定しました。OmniSci は、タクシー乗車ベンチマークのすべてのクエリーで、Spark* クラスターを大幅に上回っています (**図 5**)。



5 **インテル® CPU で OmniSci を使用したタクシー乗車ベンチマーク。x 軸は各クエリーの完了時間 (秒) を示す。このベンチマークの詳細はこちら (英語) を参照。**

チームは、最初にデータをインポートすることなく、OmniSci のインメモリーデータ取り込みに使用できるハイパフォーマンスな Arrow* ベースの取り込みコンポーネントも提供しました。目標は、OmniSci のストレージシステムを使用せずに、CSV や Apache Parquet* などのスタティック・ファイルからデータフレームを作成するのに似たワークフローをサポートすることです。

現在は、エンジンのスケールとパフォーマンスを活用しながら Python* ベースのデータ・サイエンス・ワークフローに簡単に組み込むことができるように、OmniSciDB エンジンをライブラリーにコンポーネント化する作業を行っています。また、OmniSci のストレージ・サブシステムを最適化して**インテル® Optane™ DC パーシステント・メモリー・モジュール** (**図 6**) を活用する作業も行っています。最初のベンチマーク結果は非常に有望であり、OmniSci がハードウェア・フットプリントを削減しつつ、非常に大規模なデータセットをサポートできる可能性が示されました。



6 **インテル® Optane™ DC パーシステント・メモリーで OmniSciDB を使用した予備スケーリングの結果**

OmniSci の高速データ・レンダリング・パイプラインを活用するため、**oneAPI** (英語) を使用して新しいハードウェア、特に**インテルの Xᵉ GPU** をターゲットにすることも検討しています。これにより、OmniSciDB エンジンだけでなく、OmniSci スタック全体を、データセンターからデスクトップやラップトップまで、あらゆるインテル® プラットフォームで実行できるようになります。

データ・サイエンティストにとってのメリット

この共同作業は、データ・サイエンティストにもれなく大きなメリットをもたらします。初めて、非常に効率的なハードウェア・フットプリントで数十億行のデータを含むデータセットに対して大規模な分析計算を実行できるようになります。このスケーラビリティー、パフォーマンス、熟知性の組み合わせにより、インテルと OmniSci の共同作業は、データレイクやデータ・ウェアハウスなどの大規模なデータ処理およびストレージシステムを補完するハイパフォーマンス・データ・サイエンス環境として魅力的です。

テラバイトまでのデータセットは、ラップトップでインタラクティブに分析および可視化できます。デスクトップ・クラスのシステムでは、最大 10TB 以上を分析できます。Arrow* を使用すると、**インテル® データ・アナリティクス・アクセラレーション・ライブラリー**などのマシンラーニング・ライブラリーをワークフローにシームレスに統合できます。

スケーラビリティー、パフォーマンス、柔軟性

OmniSci とインテルはともに、データ・サイエンティストに魅力的な新しいプラットフォームを提供します。オープンな、ハードウェアを考慮した、ハイパフォーマンスなアナリティクス・エンジンの機能とインテル® テクノロジー・エコシステムの能力を統合することにより、データ・サイエンス・ワークフローのスケーラビリティー、パフォーマンス、柔軟性において大きなメリットが得られることが示されています。

手順 (英語) に従って、Docker* を実行している Mac* または Linux* ラップトップで OmniSciDB をダウンロードして試すことができます。OmniSci について学び、データ・サイエンス・ワークフローを構築するのに役立つ関連情報を次に示します。

- **データサイエンスにおける OmniSci の利用についてのブログ (英語)**
- **データサイエンスにおける OmniSci のオンデマンド・セミナー (英語)**
- **Todd Mostak 氏による OmniSci アーキテクチャーの説明 (英語)**