AI-Stack

AI開発のボトルネックを突破 GPU自動スケジューリングで計算リソースの利用率を最大化

AI-Stack：次世代AIインフラ管理&AI開発・デプロイプラットフォーム

2025-02-24

はじめに 3

深層学習が企業と出会うとき 4

データサイエンスのブレークスルー 4

深層学習はソフトウェアエンジニアリングとは異なる 5

IT とデータサイエンスの完全統合はまだ実現していない 7

Kubernetes とは？AI とデータサイエンスにおける役割 8

AI-Stack：GPU リソースのスケジューリングと AI インフラ管理を簡素化 10

データサイエンスワークフローの 3 つの段階 11

AI-Stack GPU リソース最適化システム 13

AI-Stack ワークロードスケジューリングシステム 15

GPU リソース割り当て方式 16

GPU リソース配分の事例 18

他の研究者が加わると、何が起こるのでしょうか？ 19

AI-Stack 実際の顧客事例 20

AI-Stack を活用して利用率を向上、実験スピードを加速 21

結論 22

はじめに

AI 技術は急速にあらゆる業界に浸透しています。音声制御デバイス、自動運転車、医薬品開発、疾病治療など、多くの分野で革新的な製品やサービスが登場し、AI を活用する企業は、製品のインテリジェンス化や業務・意思決定プロセスの最適化を加速させています。こうした変化は、業界全体だけでなく、企業の運営モデルにも大きな影響を及ぼしています。

この変革の中心にあるのが深層学習（Deep Learning）です。深層学習は機械学習（Machine Learning）の一分野であり、人間の脳の神経回路を模倣した複雑なニューラルネットワークモデルに基づいています。こうしたモデルの開発には膨大な計算リソースが必要となるため、新型ハードウェアアクセラレータ（GPU や TPU など）が計算負荷を支える鍵となります。

多くの企業は、AI 計算リソースへの投資を拡大し、オンプレミスのデータセンターやクラウドサービスを活用しています。しかし、コスト効率よくこれらの貴重な計算リソースを管理することは、多くの企業にとって大きな課題となっています。

AI-Stack は、高コストかつ限られた計算資源という課題を解決します。コンテナ技術、高性能コンピューティング（HPC）、分散処理の仕組みを巧みに統合し、深層学習分野に応用。パフォーマンスとコスト効率の最適なバランスを実現しています。

深層学習が企業と出会うとき

では、どの業界が AI を積極的に導入しているのでしょうか？AI は企業の成長を加速させ、半導体、エネルギー、製造、交通、医薬、リテール、自動車、金融、さらには研究機関や政府機関に至るまで、幅広い分野に影響を与えています。

Statista の調査レポートによると、2024 年における AI の導入は世界中の企業で大幅に増加しました。少なくとも 1 つの業務機能に AI を統合している企業の割合は 72%に達し、2023 年の 55%から急増。特に注目すべきは、生成 AI の爆発的な成長です。全世界で 65%の企業がすでに生成 AI を導入しており、新興技術としての段階を超えて、本格的なビジネスツールとしての地位を確立しつつあります。

データサイエンスのブレークスルー

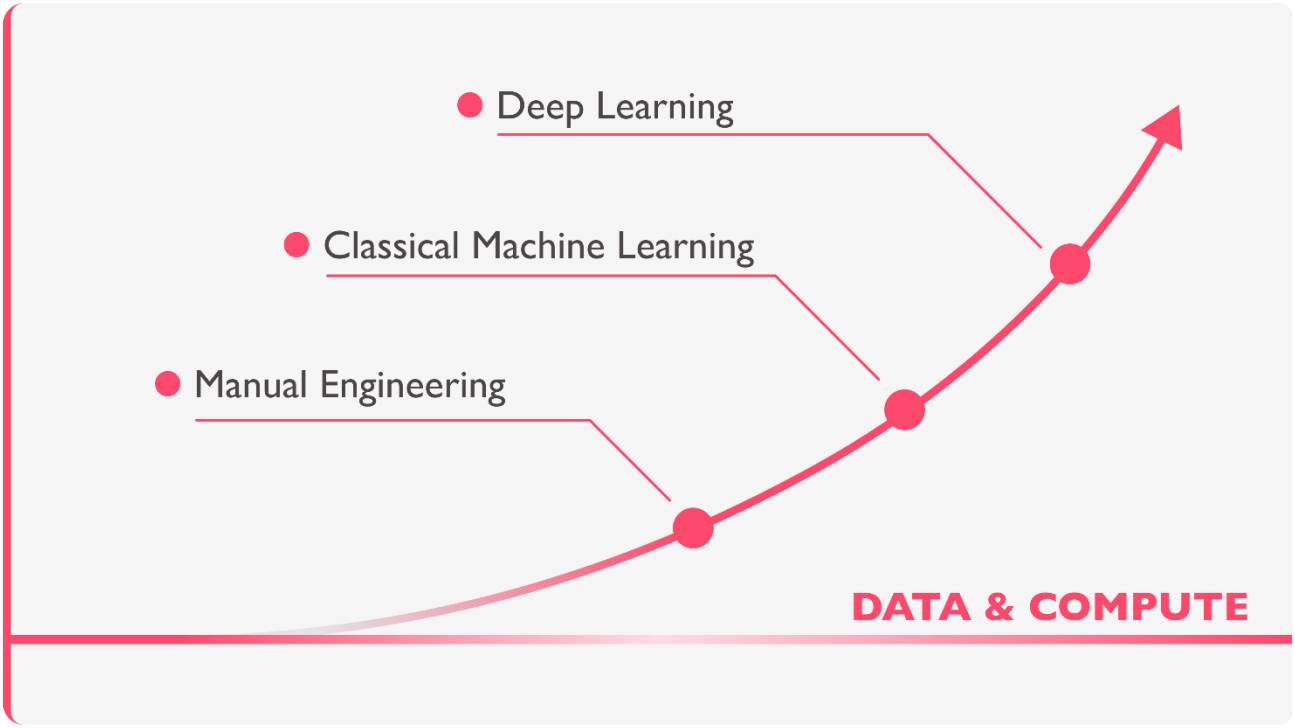

図 1：深層学習を支えるデータと計算需要の指数関数的な増加

深層学習の発展を支える 2 つの主要要因：

- **企業のデータ量の急増**：テキスト、画像、動画などのデータが爆発的に増加し、深層学習モデルの学習を加速。これにより、自然言語処理（NLP）やコンピュータビジョン（CV）といった分野の進歩が推進されています。

- **計算能力の指数関数的な向上**：NVIDIA GPU や Google TPU などの高性能ハードウェアアクセラレーターの普及により、企業は従来よりも容易に強力な計算リソースを確保できるようになり、AI アプリケーションの導入スピードが飛躍的に向上しています。

このような要因により、深層学習の企業導入は急速に拡大しています。

多くの企業が、深層学習技術の成功事例を目の当たりにし、特定の市場や消費者向けの革新的な AI 技術の開発に積極的に取り組むようになっています。しかし、データサイエンスの導入には依然として多くの課題があり、適切な計画がないと期待する成果を十分に得られない企業も少なくありません。

深層学習はソフトウェアエンジニアリングとは異なる

企業の IT 部門は、すでに従来のソフトウェア開発ライフサイクルに適応しており、近年では DevOps の普及によって開発プロセスが自動化され、デプロイの効率とスケールが向上しています。DevOps は、アプリケーションの構築やデプロイを合理化しますが、AI、特に深層学習や機械学習モデルの開発とは大きく異なる性質を持っています。

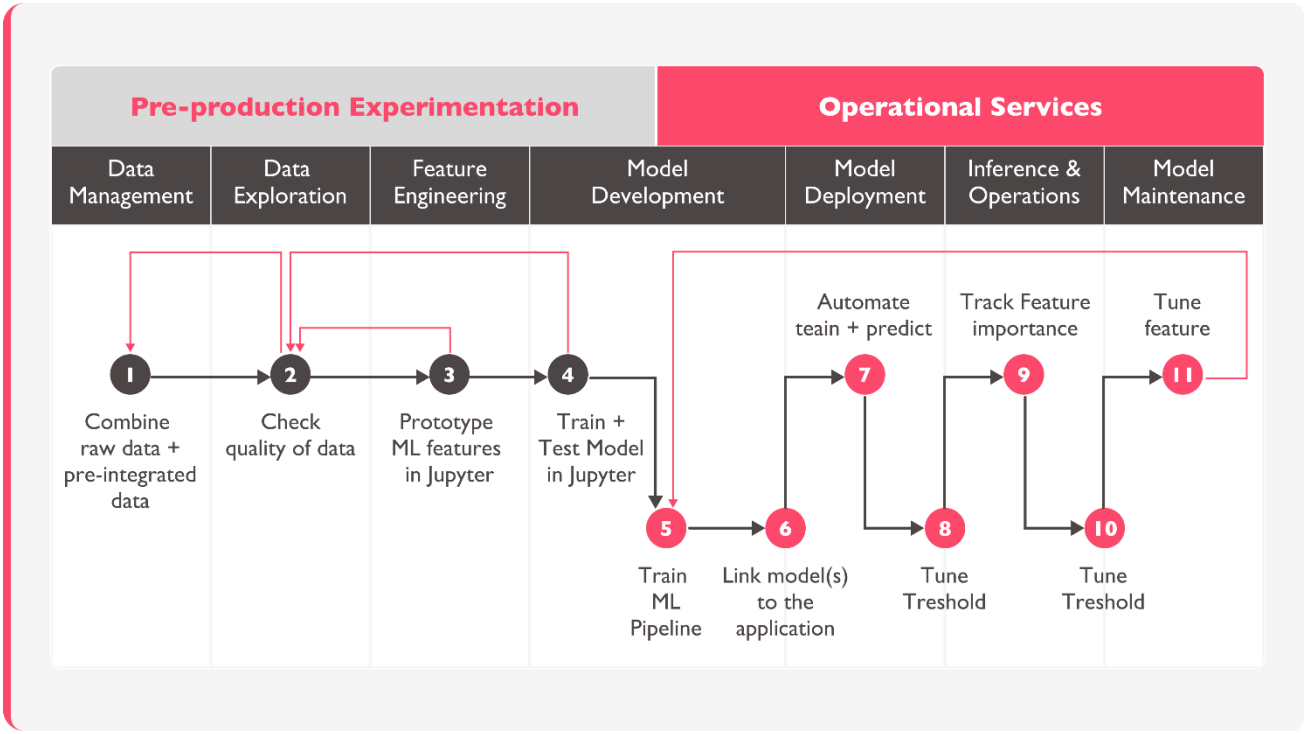

図 2：機械学習・深層学習モデルの開発プロセスとソフトウェア開発の違い

深層学習は機械学習の一分野であり、従来のソフトウェア開発とは異なり、実験的なアプローチが求められるのが特徴です。開発には頻繁なイテレーションが必要であり、モデル設計、課題の探求、データの準備、モデルのトレーニングといった複雑なプロセスを経た後、最終的に本番環境へデプロイされます。このプロセスの多くは、今後さらに自動化が進むと予想されますが、現時点ではデータサイエンティストによる手作業に依存する部分が多いのが現状ですソフトウェアエンジニアリングと深層学習の最大の違いの一つは、計算インフラへの要求の違いです。

深層学習の開発は実験から始まり、最適なモデルパラメータを見つけることがその核心となります。つまり、特定の課題に対して最適なモデルアーキテクチャを選定し、ハイパーパラメータの調整、重み付け、最適な値の組み合わせを探求するプロセスです。

この作業は数週間にわたることもあり、通常は複数の実験が並行して進行します。そのため、データセンターにはかつてないほどの負荷がかかり、計算負荷の高いワークロードを同時に処理できる柔軟かつ高性能なインフラ環境が求められています。

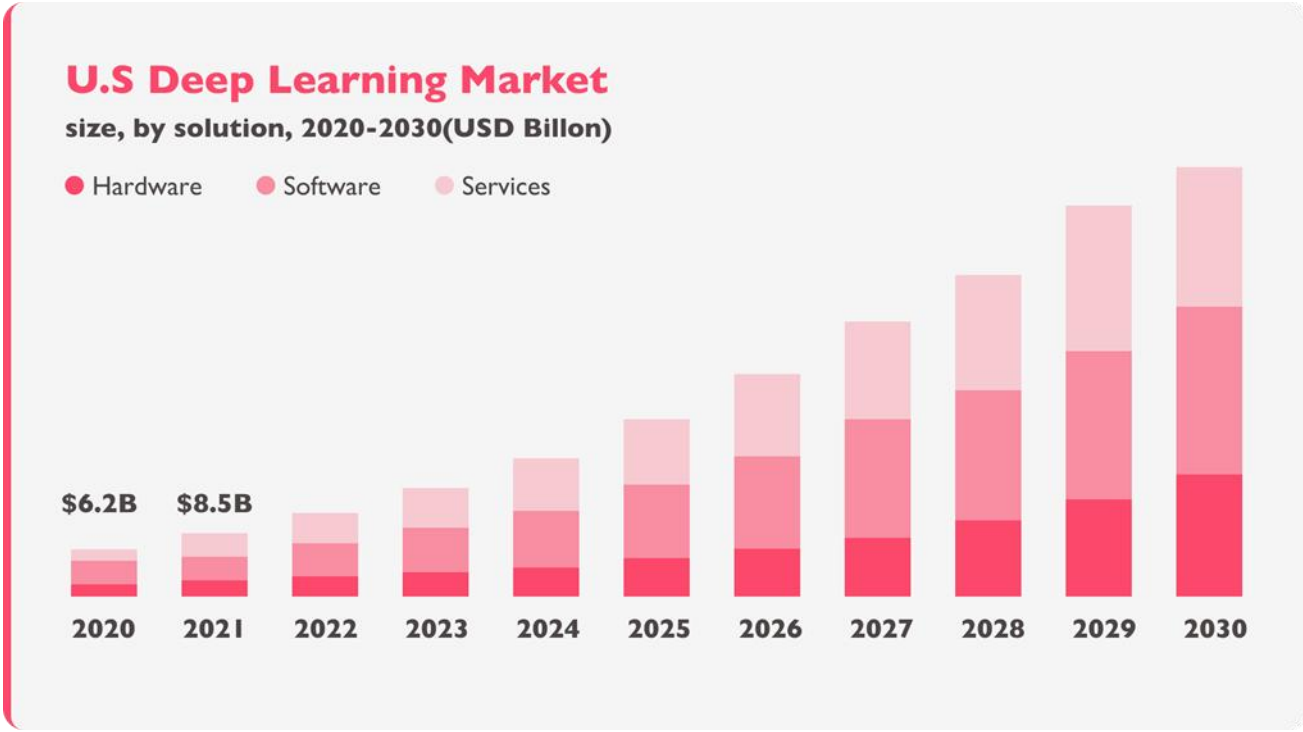

図 3：拡大するハードウェア需要が深層学習の成長を支える

(出典: Grand View Research)

これらの計算負荷の高いワークロードは専用ハードウェア上で実行されており、その普及が急速に進んでいます。上記の図が示すように、企業はこれまでにない規模で深層学習ソフトウェア、サービス、AI 専用ハードウェアに投資しています。Grand View Research の

データによると、世界の深層学習市場は 2030 年までに 5,267 億ドルに達し、2024 年から 2030 年の間に年平均成長率（CAGR）33.5%を維持すると予測されています。

AI 専用ハードウェア市場において、現在 NVIDIA の GPU がトップの座を占めていますが、競争環境は急速に変化しています。Google の TPU をはじめ、AMD、Intel、さらには Cerebras や Graphcore などの新興企業が、AI ワークロード向けに最適化された専用チップの開発を加速させています。これらの AI 専用チップは、GPU と比較して特定の深層学習タスクにおいて高いパフォーマンスを発揮する可能性があり、ハードウェアレベルでの最適化が施されています。このように AI 専用チップの普及が進むことで、計算リソースのコストが徐々に低下し、市場の需要により的確に応えられるようになると考えられます。

しかし、計算能力のコストは、企業のデータサイエンスチームが直面する課題のほんの一部に過ぎません。

IT とデータサイエンスの完全統合はまだ実現していない

機械学習プロジェクトの進展に伴い、企業はデータサイエンスと実験的アプローチを活用し、課題を解決することの重要性をますます認識しています。Precedence Research の調査レポートによると、世界の AI インフラ市場規模は 2025 年に 602.3 億ドルに達し、2034 年には 4,993.3 億ドルまで拡大すると予測されています。2025 年から 2034 年までの年平均成長率（CAGR）は 26.60%と見込まれています。

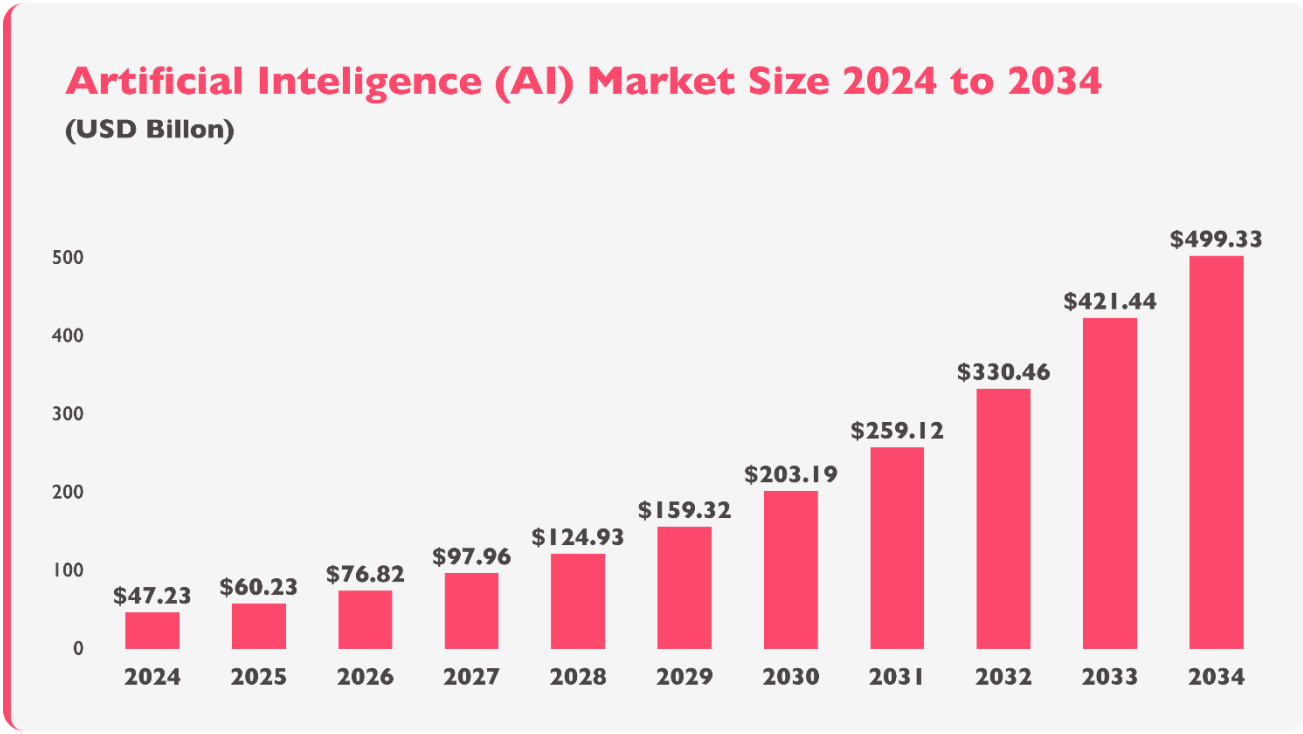

図 4：AI インフラ市場の規模は年々拡大

(出典: Precedence Research)

企業が AI インフラへの投資を拡大し、AI アプリケーションの本格導入を進める中、IT 部門の管理・運用負担が急激に増大しています。特に NVIDIA GPU などの AI アクセラレーターの導入が進むにつれ、リソースの調整・運用・スケーリングに関する課題が浮き彫りになっています。

現在、多くの GPU はデータセンター内にベアメタル環境で導入され、固定的にデータサイエンティストに割り当てられています。しかし、この静的なリソース配分方式では、AI 開発の効率を大きく損なう可能性があります。GPU リソースが柔軟に割り当てられないため、高負荷なタスクには計算資源が不足し、一方で軽量なタスクには過剰な計算資源が割り当てられるという状況が発生します。こうしたリソース配分の不均衡が、AI モデル学習におけるボトルネックを生み、AI 開発の進行を妨げる要因となっています。

451 Research の調査によると、「約半数の企業が、自社の AI インフラが現在の需要を満たしていない」と回答しています。AI の発展に伴い、単なる計算資源の確保だけでなく、GPU の管理も IT チームにとって大きな課題となっています。特に、異なるチームや拠点を跨いだインフラの可視化、ハードウェアのメンテナンス、追加の計算資源のプロビジョニングといったタスクは、より複雑化しています。

多くの GPU サーバーを導入した企業は、次第にハードウェアの管理が難しく、投資したリソースの効率が最大化できていないことに気付き始めています。その結果、AI アクセラレータクラスタの利用率が低下し、インフラの非効率性、開発プロセスの手作業依存、長期化する開発サイクルが生産性の制約となり、最終的にはユーザーエクスペリエンスの低下にもつながっています。

Kubernetes とは？AI とデータサイエンスにおける役割

ここで一度、コンテナ技術と Kubernetes について整理しておきましょう。コンテナ化技術は、現代のデータサイエンス領域において重要な役割を果たしています。コンテナは、必要なソフトウェア依存関係を含む完全な実行環境を提供し、データサイエンスや AI の実験をよりスムーズかつ効率的に進めることを可能にします。そして、Kubernetes は、次世代の IT インフラ管理ソフトウェアとして、コンテナ化されたアプリケーションを管理・運用するために設計されたプラットフォームです。Kubernetes の登場は、かつての VMware が仮想化技術にもたらした影響に匹敵し、現在ではコンテナオーケストレーションの業界標準となっています。Kubernetes を活用することで、企業はコンテナワークロー

ドのデプロイ、スケール、管理を自動化でき、IT インフラの運用効率を大幅に向上させることができます。現在、多くの企業が AI やデータサイエンスのワークロードをより柔軟に管理するための基盤として、Kubernetes を積極的に採用しています。

それでは、Kubernetes はインフラの非効率性を解決し、AI ワークフローを調整できるのでしょうか？そのスケジューラー（Scheduler）を利用すれば、AI ワークロードを効果的に管理できるのでしょうか？残念ながら、これらの問いの答えは **「いいえ」** です。Kubernetes は、AI ワークロードの運用や管理に関わる複雑な問題を完全には解決できません。

Kubernetes には、AI のスケール拡張に必要な重要なバッチスケジューリング機能が欠けています。例えば、以下のような機能が不足しています。

- **公平スケジューリング（Fair Scheduling）：**優先度やポリシーに基づいて複数のジョブキューを動的に管理し、計算資源を適切に配分する機能。
- **ギャングスケジューリング（Gang Scheduling）**：複数のノードに分散したワークロードを一括管理し、必要な計算資源を同時に確保することで、一部の計算資源のアイドル状態や計算の中断を防ぐ機能。
- **その他の AI 向けスケジューリング機能：**計算資源の利用効率を最適化するための、AI 特化の調整機能。

さらに、Kubernetes は Topology-aware scheduling（トポロジー認識スケジューリング） をサポートしていません。これは深層学習ワークロードのパフォーマンスに大きな影響を与える要素です。Topology-aware scheduling とは、スケジューラーがハードウェアの構造（CPU トポロジー、ネットワークインターフェース、GPU の相互接続など）を考慮し、計算資源の割り当てを最適化することで、処理効率を向上させる技術です。

つまり、Kubernetes のスケジューリング機能は AI ワークロード向けに設計されておらず、AI 計算の最適なスケジューラーとなるには、まだ長い道のりがあります。そのため、Kubernetes だけに依存するのではなく、AI ワークロードに特化した資源調整と管理ソリューションを組み合わせる必要があります。これにより、GPU の利用率を向上させ、計算効率を最大化できます。

INFINITIX Inc. の主力製品 「AI-Stack」 では、GPU 計算資源の管理・最適化と、Kubernetes を活用したシステム導入の簡素化という 2 つの主要課題に対応する包括的なソリューションを提供しています。次の章では、これらの機能と利点について詳しく解説します。

AI-Stack：GPU リソースのスケジューリングと AI インフラ管理を簡素化

INFINITIX の主力製品「AI-Stack」は、データサイエンティストや深層学習エンジニアの開発プロセスからハードウェアアクセラレーションリソースを抽象化し、リソースの動的割り当てと最適化管理を実現する ことをコア設計理念としています。Kubernetes ベースのアーキテクチャに基づき、「AI-Stack」は AI ワークロードと基盤ハードウェアを効果的に分離し、研究者がインフラの設定やメンテナンスに煩わされることなく、モデルの開発やトレーニングに専念できる環境を提供します。

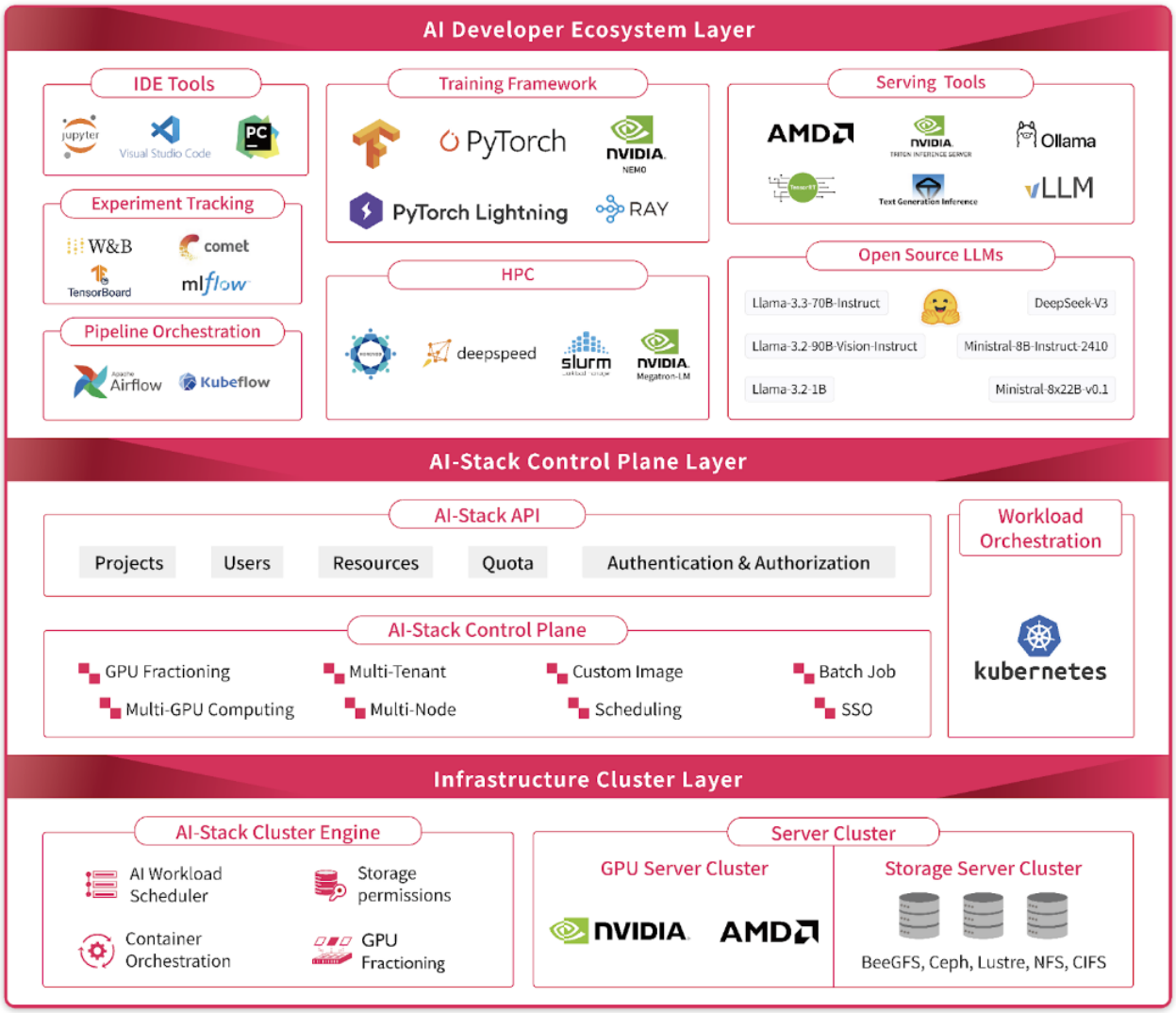

図 5：AI-Stack アーキテクチャ

また、自動化されたリソーススケジューリングと管理メカニズムを通じて、「AI-Stack」は AI モデルを適切な計算資源に効率的にデプロイし、AI 研究開発およびアプリケーションの全体的なパフォーマンスを向上させます。

「AI-Stack」 が提供する理想的なデータサイエンス計算インフラ：

- **柔軟な計算資源**：各データサイエンティストに必要な計算資源を提供。
- **自動スケジューリング処理**：ワンクリックでモデル学習や大規模分散計算を実行。
- **GPU 計算資源の最適化**：GPU の効率的な活用と拡張性の高いツールを提供。
- **Kubernetes との統合**：シンプルなプラグインで、データサイエンティストや IT チームのシステム管理負担を軽減。

「AI-Stack」 は自動化分散計算技術を備えており、IT 管理者が計算資源をより正確に割り当て、GPU のアイドル時間を削減し、クラスタの利用率を向上させることを可能にします。データサイエンティストにとって、コンテナ化された AI インフラは以下のメリットを提供します。

- **より多くの GPU 計算資源を活用し、**より多くの実験を実行することで、生産性と研究の質を向上。
- **複数の GPU による分散学習を実施し**、モデルの学習時間を大幅に短縮。

これらの最適化により、ハードウェアの利用率を少なくとも 2 倍向上させ、モデルの学習速度を 2 倍以上加速。AI インフラの運用効率と投資対効果を最大化します。

データサイエンスワークフローの 3 つの段階

AI-Stack は、データサイエンスのワークフローをモデル構築（Build）、モデル学習（Train）、モデル推論（Inference）の 3 つの異なるプロセスに分類します。一部の組織ではこれらを単一のワークフローとして扱うこともありますが、計算の観点から見ると、それぞれのニーズや特性は大きく異なります。

1. **モデル構築（Build）**
   モデル構築の段階では、データサイエンティストがモデルを開発、調整、テストし、データとコードが正常に連携することを確認します。このフェーズでは、研究者・コード・データ・ハードウェアリソースの継続的な相互作用が必要であり、試行錯誤とリアルタイム調整を繰り返します。計算負荷は比較的軽く、GPU のワーク

ロードは短周期かつ低利用率であるため、高性能計算の要求はそれほど高くありません。

2. **モデル学習（Train）**
   モデル学習フェーズでは、膨大な計算リソースが必要となり、パラメータの調整や最適化に数日から数週間を要することもあります。このワークロードは特に GPU の計算能力とメモリ性能に強く依存するため、GPU リソースの最適化が不可欠です。効率的なリソース管理によって、計算性能とコストパフォーマンスを最大限に引き上げることが重要です。

3. **モデル推論（Inference）**
   モデル推論フェーズでは、学習済みのモデルが実際のアプリケーションにデプロイされ、ユーザーリクエストに応じた処理を行います。この段階では、GPU リソースの需要が状況により大きく変動し、トラフィック負荷も時間帯によって変わるのが特徴です。そのため、推論フェーズでは、モデルの要求に応じて動的に GPU リソースを割り当てることが鍵となります。必要に応じて GPU を部分的に割り当てる、または複数 GPU で並列処理を行う ことで、パフォーマンスを最適化できます。さらに、トラフィックのピーク時には自動スケールアウト（モデルの拡張デプロイ）を行い、負荷が低下した際にはリソースを縮小し、効率的な運用が重要です。

データサイエンスの各段階における作業特性、ワークロードおよび GPU 配置の違いは以下の表にまとめられます。

モデル構築（Build）	モデル学習（Train）	モデル推論（Inference）
開発・デバッグ	モデル学習、パラメータ最適化	モデルサービスの実運用
インタラクティブな実行	自動実行	サービスのトラフィックに応じて実行
短周期実行	長時間実行	実行時間が変動
スループットはそれほど重要ではない	スループットは非常に重要	スループットの変動が激しい
GPU 利用率低い	GPU 利用率高い	GPU 利用率が激しく変動

表 1：データサイエンスワークフローの 3 つの段階

これら 3 つのワークフローの違いは明確に示されました。次に、AI-Stack プラットフォームの AI モデル開発ライフサイクルを背景に、それぞれのワークフローの適用方法について詳しく探っていきます。

AI-Stack GPU リソース最適化システム

技術の進歩により、単一 GPU の計算能力とメモリ容量は大幅に向上しています。NVIDIA GPU のメモリ容量は、初期の 1.5GB から最新世代では 192GB へと進化し、GPU コア数の増加とともに計算性能も飛躍的に向上しました。

この大容量メモリと高コア数の構成は、モデル深層学習などの高負荷な計算処理には不可欠ですが、一方で、モデル推論などの軽量なワークロードには過剰な計算資源となる場合があります。そのため、ワークロードに応じて GPU を適切に分割し、単一 GPU を複数のタスクへ柔軟に割り当てることで、GPU の利用率を最大化し、計算資源の最適化と高効率運用を実現できます。

AI-Stack は、Kubernetes 上のコンテナ化ワークロード向けに最適な GPU リソース管理システムを提供します。このシステムは CUDA ベースのワークロードに対応し、特にモデル推論やモデル構築などの軽量 AI タスクに適した GPU 最適化環境を提供します。AI-Stack の GPU 最適化システムを活用することで、データサイエンスチームや AI エンジニアリングチームは 1 枚の GPU 上で複数のワークロードを同時に実行可能になり、同じハードウェアでより多くのタスク（コンピュータビジョン、音声認識、自然言語処理など）を実行できるようになります。これにより、企業は運用コストを大幅に削減できます。

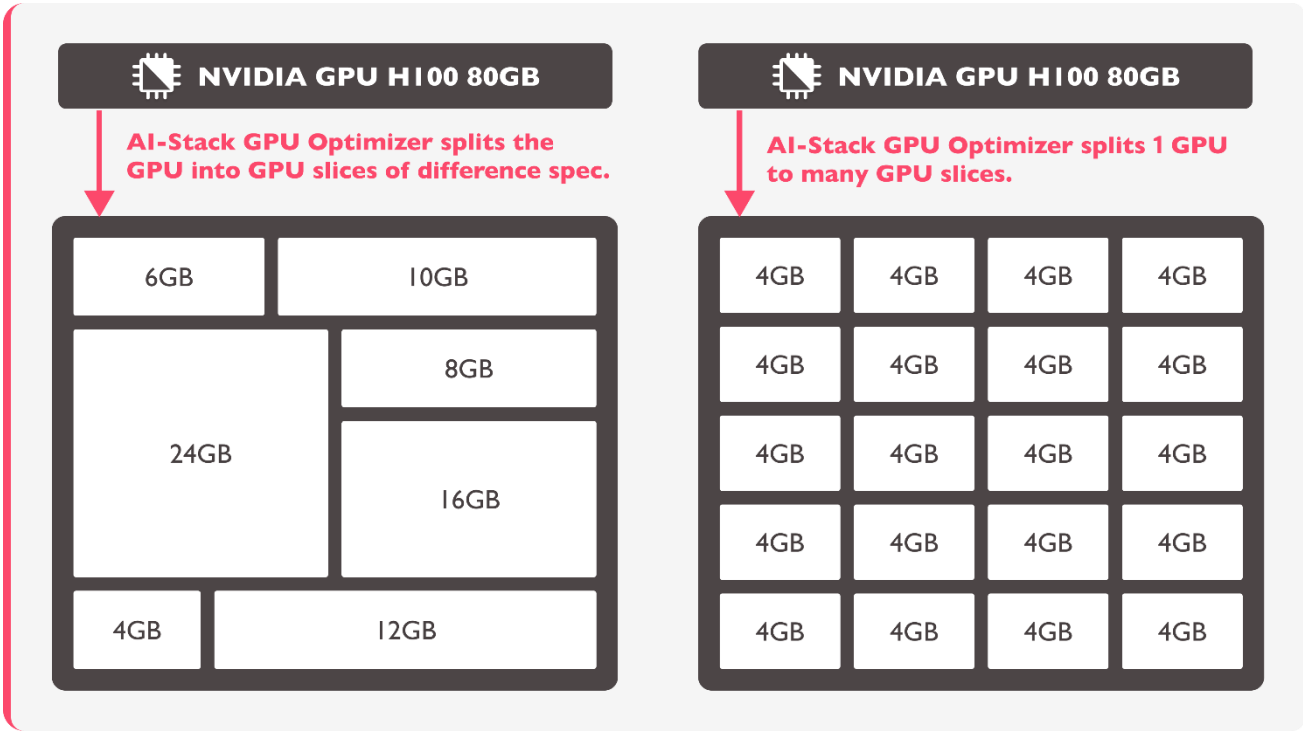

図 6：AI-Stack がワークロードに応じて 1 枚の GPU を複数の論理 GPU に分割

さらに、AI-Stack の GPU リソース最適化システムは、単一 GPU 上に異なる仕様の仮想 GPU（論理 GPU）を複数作成することが可能です。それぞれの仮想 GPU は 独立したメモリ領域と計算スペースを持ち、コンテナごとに分離してアクセス・実行できるため、物理 GPU と同様に安定したパフォーマンスを発揮します。これにより、複数のワークロードが 1 枚の GPU 上で並行して動作しつつ、互いに干渉しない環境を実現します。

AI-Stack の GPU リソース最適化システムは、透明性が高く、シンプルかつ移植性に優れており、コンテナ自体に変更を加えることなく導入可能です。標準的なユースケースでは、単一 GPU 上で 10 以上のワークロードを同時に処理し、GPU の利用効率を 10 倍以上に向上させることが可能です。例えば、NVIDIA H100 80GB などのハイエンド GPU を活用すれば、最小 1GB の単位で GPU を分割でき、最大 80 個の仮想 GPU を作成し、同時に 80 個のコンテナを並行実行することも可能です。これにより、計算資源の柔軟性と効率性が飛躍的に向上します。

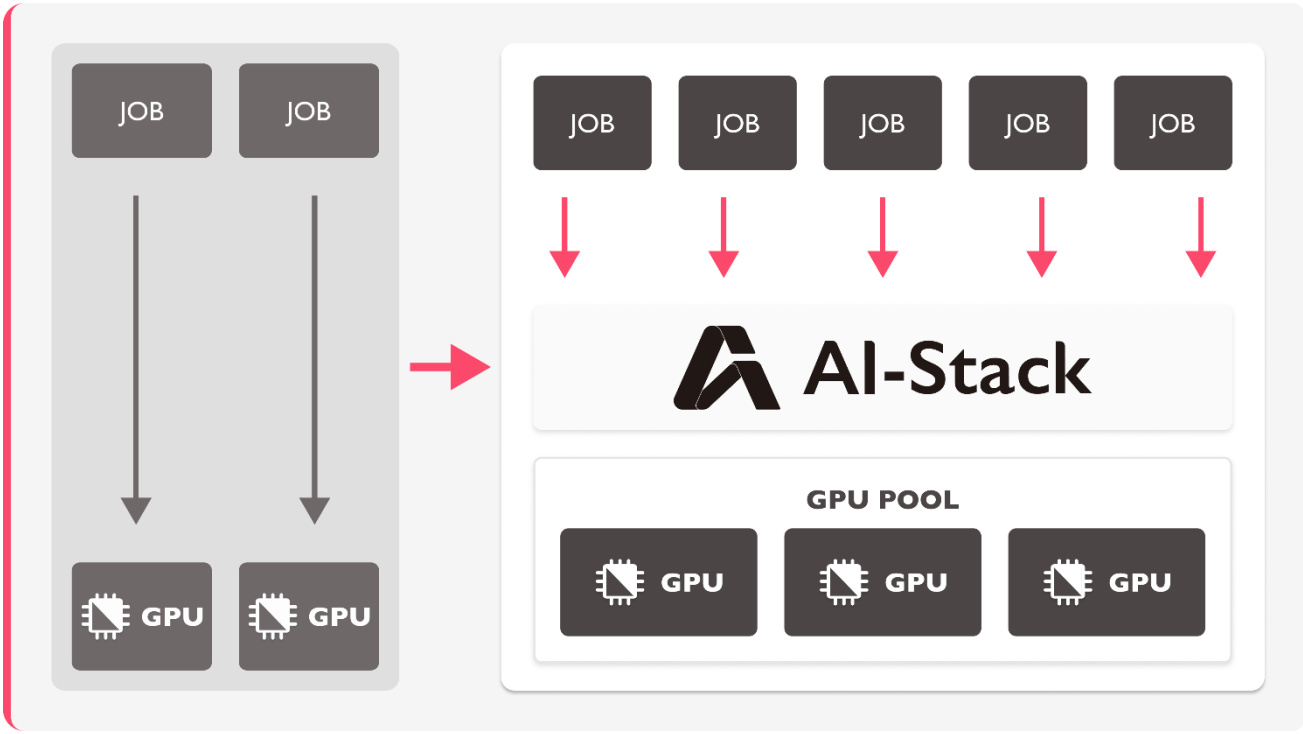

図 7：AI-Stack が異種リソースをプール化し、リソース利用率を向上

AI-Stack の GPU 分割機能 は、モデル構築やモデル推論などの軽量ながら大量の AI タスクに特化しており、データサイエンティストや AI エンジニアリングチームのモデル開発・デプロイのニーズに対応します。

異種リソースを統合することで、複数の論理環境間で柔軟なスケジューリングを実現し、モデル構築・モデル学習・モデル推論といった各フェーズの計算特性に最適化されたリソース活用を可能にします。この仮想リソースプールは、Kubernetes クラスタ内に直接デプロイでき、リソースの統合管理と高効率な運用を実現します。

AI-Stack ワークロードスケジューリングシステム

AI-Stack の独自ワークロードスケジューリングシステム は、Kubernetes 上で動作し、深層学習ワークロードの管理を最適化します。主な機能として、高度なマルチキュー管理、固定・保証リソースクォータ、優先度・ポリシー制御、自動一時停止・再開、マルチノードトレーニング対応などが含まれており、複雑なスケジューリングプロセスをシンプルかつ効率的に管理できます。

AI-Stack のスケジューラーを活用し、リソースプールを最適に管理することで、システム管理者はリソース配分をビジネス目標に合わせて柔軟に調整でき、ユーザーの追加やハードウェアの保守・拡張も容易になります。また、GPU の使用状況や利用率の可視化が可能となり、運用の最適化に貢献します。一方で、データサイエンティストは IT 部門に依存せず、必要なリソースを自動的に割り当てられるため、研究・開発の生産性を大幅に向上させることができます。

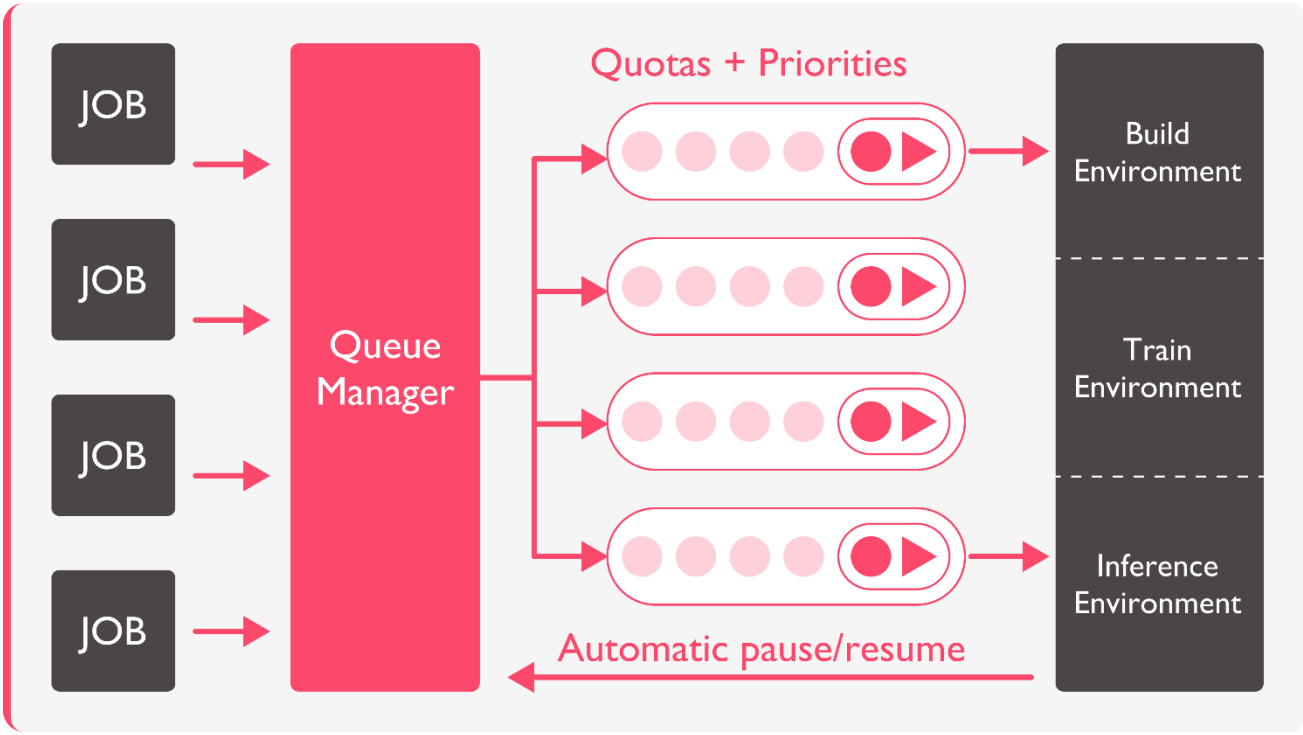

図 8：独自の AI-Stack ワークロードスケジューリングシステム

複数の論理リソースプールと AI-Stack のワークロードスケジューリングシステムが連携し、モデル構築・学習・推論のワークロードを効率的に管理します。

GPU リソース割り当て方式

現在、多くの深層学習プロジェクトでは、固定リソース割り当て方式が採用されています。これは、研究者がプロジェクトごとに特定の GPU リソースを事前に割り当てる方式で、「固定割り当て」と呼ばれます。これに対し、「超過割り当て」方式は、プロジェクトが必要に応じて事前の割り当てを超えて GPU を利用できる仕組みであり、リソースの最大活用とアイドル時間の削減を可能にします。

「超過割り当て」の方式では、システムが現在の利用可能なリソースに基づいて GPU を自動割り当てし、プロジェクトが事前の割り当てを超えて GPU を使用できるようになります。この柔軟な仕組みにより、従来の「固定割り当て」の制約を打破し、データサイエンティストはより多くの並列実験を実行できるほか、マルチ GPU 学習で高いパフォーマンスを発揮できるようになります。その結果、GPU クラスタ全体の利用率が大幅に向上します。

固定割り当て方式	超過割り当て方式
主にモデル構築や推論のワークロードに適用	モデル訓練向けのワークロードに適用
GPU は常に確保され、即時利用が可能	ユーザーは事前の割り当てを超えて GPU を利用可能
	より多くの並列実験を実行可能
	より多くのモデル学習を実行可能

表 2：GPU リソース割り当て方式

理想的な状況では、「モデル構築」ワークフローで使用する GPU は常にデータサイエンティストが利用できる状態であることが望ましい。これにより、モデル開発やデバッグ環境がいつでも使用可能となり、スムーズな開発が維持されます。一方、「モデル学習」ワークフローでは、実験のために十分な計算資源が必要となります。実験段階では、モデルのニーズに応じた計算資源が求められますが、GPU がアイドル状態になった際には、システムがそのリソースを動的に他の実験に割り当てることで、全体のリソース利用率を向上させます。

各プロジェクトは組織の要件に応じて AI-Stack のスケジューリングシステムへキュー方式で送信 されます。キューは個人、チーム、または特定の業務活動単位で分類でき、それ

ぞれに優先度やリソース割り当ての設定が可能です。これにより、計算資源の配分が方式や運用目標に適合する形で管理されます。

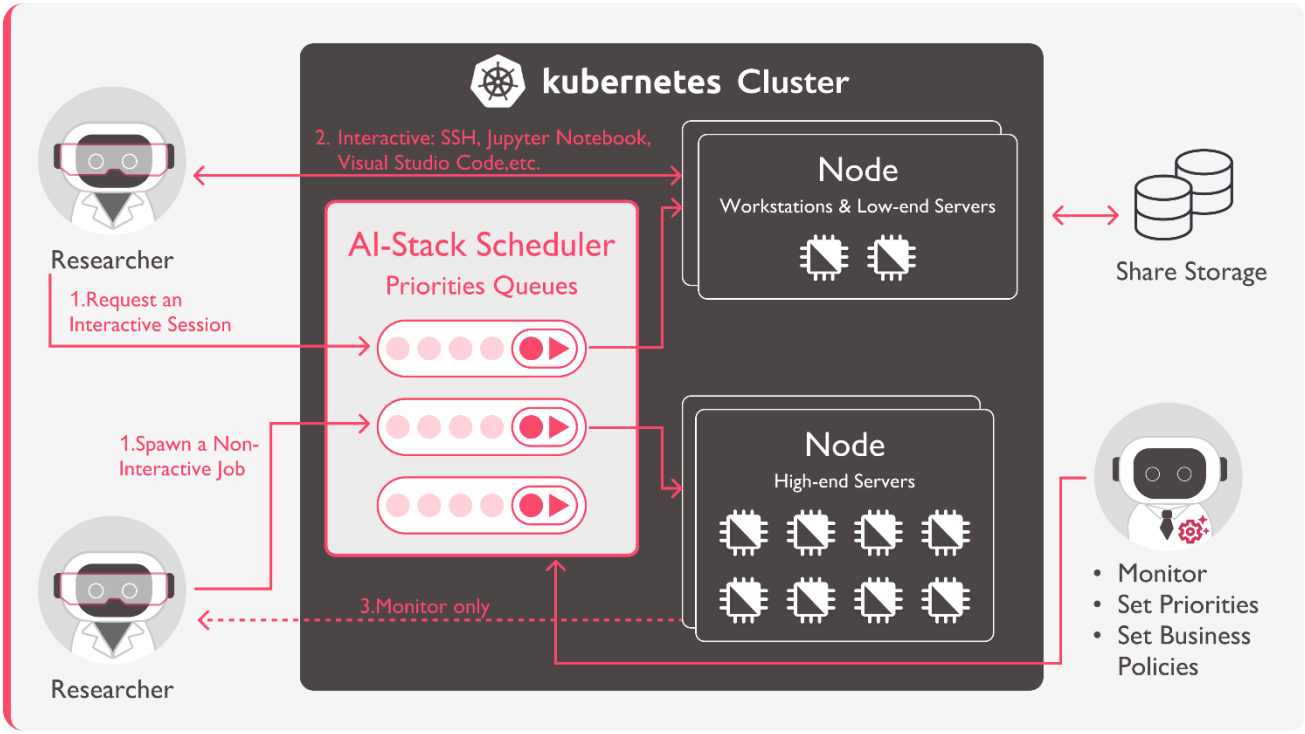

図 9：モデル構築・学習環境における GPU の研究者別割り当て

AI-Stack のスケジューリングシステムと超過割り当て方式はどのようにリソース利用を最適化するのか？図 9 で説明します。図 9 では、研究者が 2 種類の計算タスクを送信しています。インタラクティブタスク（Interactive Session）と非インタラクティブタスク（Non-Interactive Job） です。これらの計算タスクは、それぞれ異なるデータサイエンスワークフローに対応しており、インタラクティブタスクはモデル構築と推論、非インタラクティブタスクはモデル学習に関連しています。

- モデル構築および推論環境では、固定割り当て方式の採用を推奨します。
  - GPU リソースは固定的に割り当てられ、他のユーザーと共有されません。常に利用可能な状態を維持し、研究者の開発およびデバッグ作業をスムーズに進められるようにします。
  - インタラクティブ実行（SSH、Jupyter Notebook など）は、GPU サーバーノードへ直接接続し、ユーザーが即時にリソースへアクセスできるようにします。

- モデル学習環境では、規模が大きいため、GPU プール化と超過割り当ての採用を推奨します。

- GPU リソースをプール化することで、異なるワークロードがすべての利用可能なリソースを柔軟に共有し、利用率を向上させます。
- ユーザーは事前の割り当てを超えた GPU を使用可能であり、高優先度のキューは GPU 割り当てを優先的に受け取ることができます。リソースが不足した場合、スケジューラーは優先度と公平性の原則に基づいて、一部の低優先度タスクを一時停止します。リソースが十分に確保できた際には、低優先度キューもアイドル状態の GPU を利用可能になります。

GPU リソース配分の事例

実際の例で説明します。データサイエンティストの Hunter は、毎日 2 つの専用 GPU という固定リソースの割り当てを受けています。これらの GPU は 彼専用で、他のユーザーと共有されていません。短期的に見ると、特にモデル構築段階では問題なく作業できており、ワークフローも順調です。彼の主な関心は、モデルの開発とデバッグにあります。

しかし、モデルトレーニングの段階に入ると状況が変わります。突如として、より多くの GPU 計算資源が必要になるのです。にもかかわらず、リソース配分の制限により追加 GPU を利用できないという問題に直面します。たとえ同僚が自分の GPU を使っていなくても、Hunter はそのリソースを使うことができません。

通常、Hunter が追加の計算資源を必要とするのは、以下のような理由からです：

1. より多くの実験を並列に実行したい。
2. マルチ GPU による高速トレーニングや大きなバッチサイズを扱うことで、モデル精度を向上させたい。
3. 1 つのモデルをトレーニングしながら、新たなモデルの構築も同時に進めたい。

しかし、AI-Stack を活用することで、Hunter はこれらの目標を問題なく達成し、より効率的なワークフローを実現できます。その鍵となるのが、超過割り当て方式です。Hunter チームのすべてのリソースは集中管理されており、これにより彼はより大きな GPU 割り当てを受け取ることができ、AI-Stack の超過割り当て方式に基づき、Hunter を含むすべてのユーザーに柔軟な計算資源が提供されます。

前述の通り、Hunter チームの各データサイエンティストには、モデル構築段階において固定のリソースが割り当てられており、開発に必要な GPU は常に使用可能な状態が確保されています。しかし、彼らがより多くの GPU を必要とする場合、たとえば、実験数を増やす、大規模なマルチ GPU トレーニングを実行する、あるいは複数の実験を同時に進

行するといった場面では、タスクをより大きな共有リソースプールへ送信するだけで、システムが自動で調整してくれます。

さらに、Hunter が自身に割り当てられた GPU を使用していない間は、そのリソースを他のユーザーが活用可能です。そして、彼が再びその GPU を必要とした際には、システムが優先的に割り当てを復元し、一時停止していたタスクは自動的に再開されます。これが、Hunter が柔軟にリソースを活用できる AI-Stack の「超過割り当て方式」 の仕組みです。

他の研究者が加わると、何が起こるのでしょうか？

以下の図では、Hunter、Allen、Evan の 3 名が 8 枚の GPU を備えたクラスタを共有しています。静的割り当てまたは固定割り当て方式の場合、それぞれのユーザーに 2 枚の GPU が事前に割り当てられます。一方、超過割り当て方式では、各ユーザーの GPU 割り当ては 仮想的に定義されます。Hunter は常に 2 枚の GPU にアクセス可能ですが、もし彼が 4 枚の GPU を必要とし、かつ他のユーザーがそのリソースを使用していなければ、システムは一時的に 4 枚へのアクセスを許可します。その後、Allen や Evan が GPU を必要とした場合、システムはそれらのリソースを回収し、再スケジューリングします。

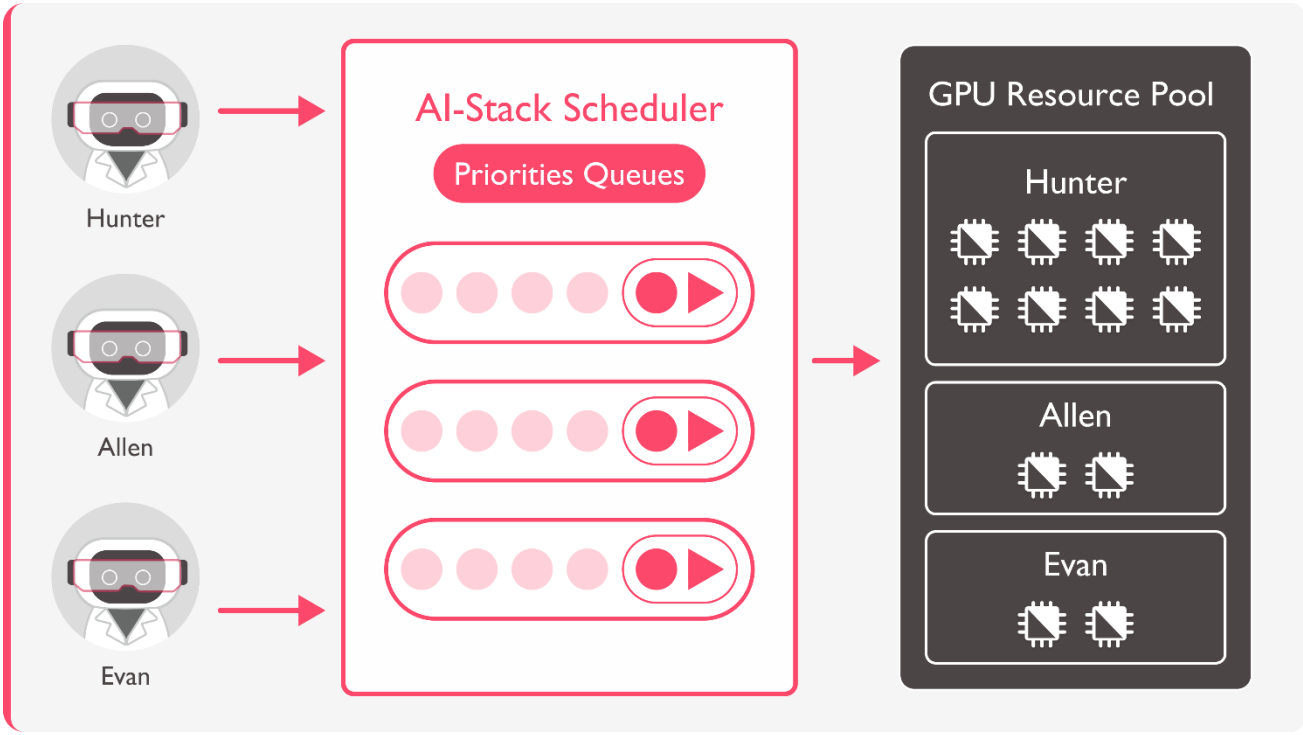

図 10：GPU リソースプールを共有してモデルをトレーニング

まとめると、超過割り当て方式では、ユーザーのリソースがアイドル状態で利用可能な場合、たとえ当初の割り当てを超えていても、システムはそのリソースをジョブに割り当てます。新たなタスクがシステムに投入されると、AI-Stack のスケジューラーはタスク目標に基づいて各キューを管理し、優先度とポリシーに応じてワークロードを一時停止または再開できる仕組みを提供します。このような方式は、クラスタ全体のリソース利用率を向上させるだけでなく、データサイエンスチームの生産性も大幅に向上させます。

AI-Stack 実際の顧客事例

ある AI-Stack の導入企業では、さまざまな種類の GPU を備えた異種ハードウェア環境でエンジニアリングチームが業務を行っていました。構成には、NVIDIA GeForce や Quadro などの低スペック GPU を搭載したワークステーションから、Tesla A100/H100 GPU や DGX サーバーなどの高性能サーバーまでが含まれていました。AI-Stack を導入する前、この企業では静的なハードウェア割り当て方式を採用していました。当初、同社のデータサイエンティストやエンジニアチームは小規模であり、各ユーザーに異なる数・種類の GPU が手動かつ固定的に割り当てられていました。

しかし、チーム規模と深層学習プロジェクトの拡大に伴い、IT 部門は需要に対応すべくハードウェアを増設し続けましたが、期待された生産性の向上は実現せず、むしろ多くの課題が顕在化しました。新たに参加したユーザーには専用のリソースが与えられても、チーム全体としての計算効率は上がらず、ハードウェア投資が増加する一方で、データサイエンティストの生産性は基盤の制約により頭打ちとなり、コストは膨らみ、投資対効果の測定が困難になっていました。

AI-Stack が当該クラスタの GPU を初期分析した結果、以下の問題点が速やかに明らかになりました：

- GPU のスペックが多様で、高性能モデルと低性能モデルが混在しており、一元的な管理が困難
- クラスタ全体の GPU 平均利用率が 30% 未満にとどまっている
- マルチ GPU を用いたトレーニングタスクの使用率は比較的高いものの、利用状況の変動が大きい
- モデル構築に関連するインタラクティブなワークロードの利用率が極めて低く、GPU は長時間アイドル状態。稼働中であっても利用率は 50% を下回っていた

このようなリソース配分のあり方では、最適化がされているとは言えず、一部のユーザーが GPU を断続的に使用している一方、大半の GPU は長時間未使用のままとなり、結果として全体の利用率が 30% を下回るという非効率な状況に陥っていました。

AI-Stack を活用して利用率を向上、実験スピードを加速

企業が AI-Stack を導入する際、最初のステップは すべての GPU ノードを AI-Stack に統合し、リソースを集中管理することです。次に、GPU は用途に応じて 2 つのグループに分類されます。

1. **固定割り当て方式のモデル構築グループ**：このグループでは、各研究者に 1 台のワークステーションと、低スペックの NVIDIA GeForce 系列 GPU を 2 枚、固定で割り当てます。研究者は、インタラクティブなモデル構築時に 2 枚の GPU を自由に使用できますが、このグループ内のリソースは 他のユーザーと共有されません。

2. **超過割り当て方式のモデル訓練グループ**：このグループは、モデルトレーニング専用に設計されています。高性能 GPU リソースは、トレーニング中の研究者に優先的に割り当てられ、NVIDIA DGX サーバーなどのハイエンド機器を活用して、AI-Stack を通じて すべての高性能 GPU を一元管理します。このグループのリソースは 全研究者で共有可能であり、数日間にわたる連続的なワークロードに対応しつつ、結果の出力スピードを加速する仕組みになっています。

このような運用体制のもとで、AI-Stack は包括的な監視機能を提供し、企業に以下のメリットをもたらします：

- すべての GPU の状態と使用状況をリアルタイムで把握
- 管理者がリソース利用率をさらに最適化
- ユーザー自身で AI 開発環境を構築可能
- AI モデルを迅速にデプロイ

AI-Stack 導入後、顧客は以下のような顕著な成果を得ています：

- **実験数が大幅に増加し、成果のスピードも加速：**GPU あたりの実験数で換算すると、生産性は 10 倍以上向上。
- **GPU の平均利用率が 90%を超え、**ハードウェアリソースの最適化を実現。
- **投資回収率（ROI）の向上**：リソースを効率的に運用することで、ハードウェアコストを削減。

- **研究者のワークフローを簡素化**：データサイエンティストは基盤やツールの管理に時間を費やすことなく、実験の質と量に集中でき、さらなるイノベーションを推進。
- **IT 部門の可視性と管理性が向上**：新規ユーザーもスムーズに環境へ適応でき、ハードウェア統合とリソース割り当ても円滑に。AI-Stack の分析機能により、将来的なリソース計画も支援。

これにより、企業の AI 研究開発プロセスは大幅に効率化され、より優れたデータサイエンス成果の実現へとつながっています。

結論

深層学習はさまざまな業界で急速に活用が広がっており、それに伴って計算能力への需要も急増しています。

データサイエンスの取り組みを拡大する企業にとって、単にハードウェアを追加購入・配備するだけでは課題は解決できません。効率的かつコスト効果の高いリソース活用戦略がなければ、非効率と不透明な投資回収率（ROI）は引き続き大きな障壁となるでしょう。

AI-Stack は、異なるワークロードに対して以下の 3 種類のタスクタイプを定義しています：小規模なモデル構築タスク、大規模なモデル学習タスク、常時稼働の推論タスク

AI-Stack は精緻なリソース配分により、小規模タスクには必要十分な GPU リソースを、大規模タスクには制限のない計算リソースを提供し、推論タスクには実運用に即した柔軟なスケジューリングを実現。これにより、データサイエンスのワークロードをハードウェアから切り離し、リソースの最大活用を可能にします。

リソースを統合し、先進的なスケジューリングメカニズムを適用することで、AI-Stack はデータサイエンスのワークフローを最適化。優先すべきプロジェクトに対する確実なリソース配分を可能にし、実験スピードを加速、結果の迅速な創出を実現。最終的に、企業の AI 計画をビジネス成果へと導きます。

AI-Stack の詳細情報はこちらをご覧ください：https://ai-stack.ai/

