

# AWS での Electronic Design Automation (EDA) ワークフローの最適化

2018 年 9 月



## 注意

本書は情報提供のみを目的としています。本書の発行時点における AWS の現行製品と慣行を表したものであり、それらは予告なく変更されることがあります。お客様は本書の情報、および AWS 製品またはサービスの利用について、独自の評価に基づき判断する責任を負います。いずれの AWS 製品またはサービスも、明示または黙示を問わずいかなる保証も伴うことなく、「現状のまま」提供されます。本書のいかなる内容も、AWS、その関係者、サプライヤー、またはライセンサーからの保証、表明、契約的責任、条件や確約を意味するものではありません。お客様に対する AWS の責任は AWS 契約によって規定されています。また、本文書は、AWS とお客様との間の契約に属するものではなく、また、当該契約が本文書によって修正されることもありません。

# 目次

要約	vi
はじめに	1
EDA の概要	1
AWS クラウドのメリット	2
生産性の向上	2
高可用性と耐久性	3
コンピューティングリソースと要件のマッチング	3
アップグレードサイクルの加速化	4
EDA ワークフローを AWS に移行するパス	5
データアクセスと転送	5
Amazon S3 に移行するデータの検討	5
前提条件	6
最初の概念実証 (POC) に推奨される EDA ツール	6
クラウド向けに最適化された従来のアーキテクチャ	7
AWS での EDA アーキテクチャの構築	8
ハイパーバイザー: Nitro、Xen	8
AMI およびオペレーティングシステム	9
コンピューティング	11
ネットワーク	15
ストレージ	15
ライセンス	24
リモートデスクトップ	26
ユーザー認証	28

オーケストレーション	28
AWS での EDA ツールの最適化	30
Amazon EC2 インスタンスタイプ	31
オペレーティングシステムの最適化	31
ネットワーキング	38
ストレージ	38
カーネル仮想メモリ	39
AWS クラウドでのセキュリティおよびガバナンス	40
データ保護とデータ主権向けに分離された環境	40
ユーザー認証	40
ネットワーク	41
データストレージと転送	42
ガバナンスとモニタリング	44
寄稿者	47
ドキュメントの改訂	47
付録 A - ストレージの最適化	48
NFS ストレージ	48
付録 B - リファレンスアーキテクチャ	50
付録 C - Linux カーネルコマンドラインの更新	52
/etc/default/grub ファイルを使ったシステムの更新	52
/boot/grub/grub.conf ファイルを使ったシステムの更新	53
カーネルラインの検証	53

# 要約

Electronic Design Automation (EDA) を使用している半導体および電子機器の企業は、AWS で利用できるコンピューティング、ストレージ、リソースをほぼ無制限に活用して製品開発ライフサイクルと市場投入までの時間を大幅に加速させることができます。このホワイトペーパーでは、EDA ワークフローの概要、EDA ツールを AWS に移行する際の推奨事項、AWS で EDA ワークロードを最適化する特定の AWS アーキテクチャコンポーネントについて紹介します。

## はじめに

半導体、集積回路 (IC)、プリント回路基板 (PCB) の設計と検証に使われるワークフロー、アプリケーション、手法は、CAE (Computer-Aided Engineering) と Electronic Design Automation (EDA) ソフトウェアが発明されて以来、ほとんど変更されていません。しかし、電子機器システムと集積回路の複雑化やジオメトリの微細化に伴い、これらのシステムを設計、テスト、検証、構築するコンピューティング処理能力とインフラストラクチャの要件は著しく増大しています。CAE、EDA と、計算機リソグラフィや寸法計測などの新しく出現したワークロードにより、次世代の電子製品で膨大な規模のコンピューティングとデータ管理への必要性が高まっています。

半導体と電子機器部門では、設計時間全体の大半が Intellectual Property (IP) コアの特性評価、フルチップ機能検証、タイミング検証など、コンポーネントの検証に費やされています。EDA サポート組織 - 半導体企業にインフラストラクチャサポートを提供する専門 IT チームは拡大し続けるサーバーファームと高性能ストレージシステムに投資して、半導体のテストおよび検証の品質向上と納期短縮を実現する必要があります。アップグレードされた新しい IC 成形技術を導入するには、ハードウェアの回帰テストを高速に完了したり設計 IP の特性を再評価したりするため、比較的短い期間に大量のコンピューティングとストレージが必要になる可能性があります。

今日の半導体企業は、アマゾン ウェブ サービス (AWS) を使用し、より高速で柔軟性に優れた CAE および EDA インフラストラクチャのデプロイメントを、レジスタ転送レベル (RTL) 設計からチップ製造工場への GDSII ファイル納品まで、完全な IC 設計ワークフローで活用しています。AWS のコンピューティング、ストレージ、上位レベルのサービスは必要に応じて必要な分だけ利用できます。パフォーマンスを重視する EDA ワークロードで一般に必要なとされる大きな先行設備投資をする必要はありません。

## EDA の概要

EDA のワークロードは、ワークフローと、特に半導体集積回路 (IC) においてマイクロエレクトロニクスを効率化できるソフトウェアツールのサポートセットで構成されます。半導体の設計と検証は、シリコンテープアウトと製造を促進して所要時間を短縮する市販またはオープンソースのツールセット (総称して EDA ソフトウェアという) に依存します。EDA は、反復の多いエンジニアリングプロセスであり、1 つの集積回路を製造するのに数か月、場合によっては数年かかる場合があります。

集積回路の複雑化に伴い、事前構成または部分的にカスタマイズされたハードウェアコンポーネント（総称して Intellectual Property (IP) コアという）の利用がさらに増大しています。これらのコアは一般的なゲートレベルのネットリストとして IP 開発者から提供され、半導体企業の社内で設計されるか、サードパーティーの IP ベンダーから購入します。IP コア自体には、設計と検証や、特定の IC 成形技術のパフォーマンスを特性評価するのに EDA ワークフローが必要です。これらの IP コアは IC 固有のカスタム設計されたコンポーネントと組み合わせて使用され、多くの場合、組み込み CPU、標準の周辺機器、I/O、カスタムのアナログまたはデジタルコンポーネントを活用する複雑なシステムオンチップ (SoC) が含まれる完全な IC を作成します。

その後、すべての IP コアとカスタムコンポーネントを含む完全な IC 自体に、チップ内の全コンポーネントのモデリング（つまりシミュレーション）など、フルチップの検証に大量の EDA 処理が必要になります。このモデリングでは HDL ソースレベルの検証、物理合成、初期検証（例：フォーマル検証などの方法を使用）などが行われ、「フロントエンド設計」と呼ばれます。物理的な実装ではフロアプラン、配置配線、タイミング分析、デザインルールチェック (DRC)、最終検証などが行われ、「バックエンド設計」と呼ばれます。バックエンド設計が完了すると、ファイルが GDSII 形式で生成されます。このファイルの作成は、歴史的な背景から「テープアウト」と呼ばれています。完了すると、このファイルはシリコンウェハーが製造される製造施設（製造工場）に送られます。この施設は半導体企業によって運営されている場合も、そうでない場合もあります。このウェハーには数千もの個別の IC が格納されており、検査後、ダイ切断されてテストされ、チップにパッケージ化されて再テストされ、高度に自動化された製造プロセスを通じて基板またはその他のシステムに組み込まれます。

半導体と電子機器のサプライチェーンでは、これらのすべての工程で、クラウドのスケラビリティによる利点を活用できます。

## AWS クラウドのメリット

EDA ワークロードを AWS への移行方法の詳細について説明する前に、AWS クラウドのクラウドコンピューティングから得られる利点について把握しておきます。

### 生産性の向上

クラウドに移行する組織は、開発の生産性と市場投入までの期間を大幅に向上できます。組織は、処理を待機するジョブの需要に合わせてコンピューティングのニーズをスケールアウトすることでそれを実現できます。AWS では、コンピューティングリソースに秒単位の請求を適用しているため、秒単位の使用量だけを支払うことによりコストを最適化できます。

水平スケーリングでは、少ないサーバーを長期間実行するのと同じコストで、多くのコンピューティングサーバー（つまり、Amazon Elastic Compute Cloud [Amazon EC2] インスタンス）を短い期間で実行できます。たとえば、消費されるコンピューティング時間は同じなので、何千もの保留ジョブを並行して実行するためにクラスターを 24 倍以上に動的に拡大すれば、48 時間の設計回帰をわずか 2 時間で完了できます。

こうした極端なレベルの並列処理は、さまざまな業界やパフォーマンスを重視するユースケースに渡って、一般的に AWS 上で運用されています。

## 高可用性と耐久性

Amazon EC2 はワールドワイドでホストされています。これらのロケーションはリージョンとアベイラビリティゾーン (AZ) で構成されます。各 AWS リージョンとは、オレゴン、バージニア、アイルランド、シンガポールなど、世界中にある個別の地理的領域です。Amazon EC2 が実行される各 AWS リージョンは、その他のリージョンから完全に分離されるように設計されています。この設計により、最大限の耐障害性と安定性が達成されます。レプリケートされるようにサービスを設定しない限り、リージョン間でリソースがレプリケートされることはありません。

AWS にはそれぞれの地理的リージョン内に複数の独立したロケーションがあり、このロケーションは「アベイラビリティゾーン」と呼ばれます。Amazon EC2 では EC2 インスタンスなどのリソースや、これらのアベイラビリティゾーンを使用したデータを複数のロケーションに配置できます。各アベイラビリティゾーンは独立していますが、同じリージョン内のアベイラビリティゾーンは低レイテンシーのリンクで接続されています。複数のリージョンと複数のアベイラビリティゾーンの両方を活用すれば、障害から保護するとともに、高いコンピューティング処理能力を必要とするワークフローも実行できるだけの十分な空き容量を確保できます。さらに、このグローバル規模の展開では、低レイテンシーのパフォーマンスが重要になる IC 設計エンジニアの近くにコンピューティングリソースを配置できます。詳細については、[AWS グローバルインフラストラクチャ](#)を参照してください。

## コンピューティングリソースと要件のマッチング

AWS では、お客様のジョブに適したコンピューティングニーズに対応できるようにインスタンスファミリーと呼ばれるさまざまな構成のハードウェアが提供されます。このインスタンスファミリーとクラウドのオンデマンドの特性により、実行する必要のあるジョブに合わせて必要なシステムを取得し、必要な時間だけ実行できます。

Amazon EC2 インスタンスはさまざまなサイズと構成で用意されています。これらの構成は、大小のメモリ使用量をどちらも必要とし、高コアカウントの最新世代プロセッサと、高 IOPS から高スループットまでのストレージ要件を必要とするジョブをサポートするように構築されています。インスタンスを作業単位に適したサイズに変更することで、総コストを低く抑えた高い EDA パフォーマンスを実現できます。要求の厳しいほんの数件のジョブ需要を満たすために、完全に構成された EDA クラスタードウェアを購入する必要はありません。代わりに、サーバーを選択してサーバーのクラスタ全体を起動し、これらのクラスタを拡張/縮小したり、特定のアプリケーションやチップ開発の特定の段階向けに各クラスタを一意に最適化できます。

たとえば、重要な IP コアの開発中などに、ゲートレベルのシミュレーションをわずか数週間で実行する状況を想定してみてください。この例では、100 台の (2,000 CPU コア以上を表す) マシンの 1 つのクラスタに特定のメモリ対コア比と特定のストレージ構成を指定しなければならない可能性があります。AWS では、このクラスタをこのタスク専用でデプロイし、シミュレーションに必要な時間のみ実行して、プロジェクトのその段階が完了するとクラスタを終了できます。

今度は、それぞれがローカルにインストールされた独自の EDA IT インフラストラクチャを使用する複数の半導体設計チームがさまざまな地理的リージョンで作業する別の状況を想定します。この地理的に分散されたエンジニアリングチームは、最新のチップ設計から生産性のメリットが得られますが、大規模な EDA インフラストラクチャを管理するうえで課題 (グローバルにライセンスされた EDA ソフトウェアを効率的に利用するなど) が生じる可能性があります。AWS を使用してこれらの地理的に切り離された IT リソースを補強するか置き換えると、スケーラブルなオンデマンドのクラスタを AWS で使用し、すべてのグローバル EDA ライセンスを少数のロケーションにプールできます。その結果、静的タイミング分析、DRC、フィジカル検証などの重要なバッチ作業を迅速に完了できます。

## アップグレードサイクルの加速化

EDA ワークロードをクラウドに移動するもう 1 つの重要な理由は、最新のプロセッサ、ストレージ、ネットワークテクノロジーへのアクセス権の取得です。一般的なオンプレミスの EDA インストールでは、複数年にわたり使用することを前提として、サーバーとストレージデバイスを選択、構成、調達、デプロイする必要があります。選択したプロセッサの世代と購入時期によっては、パフォーマンスを重視する本番稼動用の EDA ワークロードが、複数のプロセッサ世代にわたりすでに何年か経過しており、古くなっているハードウェアデバイス上で実行されている可能性があります。AWS を使用することで、最新のプロセッサ世代を選択してデプロイできます。EDA ワークフローで各アプリケーションに固有のニーズに合わせて、EDA クラスタを数分で構成します。

## EDA ワークフローを AWS に移行するパス

AWS への EDA ワークフローの移行を開始すると、複数のデータセンターにまたがって従来の EDA デプロイが多数、並行管理されていることがわかります。

半導体業界の大規模な組織では一般的に設計チームが分散されているため、地理的に分離された複数のデータセンターがあります。これらの組織は通常、特定のワークロードを特定のロケーションで実行するように選択するか、グローバルな EDA ワークフローの大規模な負荷に対応するために複数のサイトでデータをレプリケートして同期できるようにします。このアプローチを使用する組織では、データレプリケーション、キャッシング、ライセンスサーバーの管理などのトピックについて、多くの内のおよび組織的な要因に応じて具体的に検討する必要があります。

複数のデータセンターに関連する同じアプローチと設計上の決定事項のほとんどがクラウドにも適用されます。AWS では既存の EDA データセンターの設計を反映する 1 つ以上の仮想データセンターを構築できます。コンピューティングリソース、ストレージサーバー、ユーザーワークステーションなどを有効にする基本的なテクノロジーが、わずかなキーストロークで利用できるようになります。ただし、EDA ワークロード向けの AWS クラウドから得られる実際の機能は、AWS が提供するダイナミックな機能と膨大なスケーリングによって実現されます。

### データアクセスと転送

クラウドでワークロードを実行することを初めて検討する場合、既存のオンプレミスのコンピューティングクラスターへの拡張としてクラウドリソースを設定するバーストシナリオを想定するかもしれません。このモデルでも正常に使用できますが、シームレスな方法でバーストをサポートするアーキテクチャを構築する際にデータの移行で大きな課題が生じます。プロジェクトごとにバーストを検討し、AWS でワークフロー全体が実行されるように選択すると、既存のオンプレミスのリソースが解放されて他のタスクを処理できるようになるため、組織は最大限のメリットが得られる可能性があります。この方法でクラウドリソースを活用すれば、AWS とデータセンター間でデータを同期しないため、はるかに単純なデータ転送メカニズムを使用できます。

### Amazon S3 に移行するデータの検討

EDA ツールを AWS に移行する前に、最初のテストから完全な本番稼働環境へ移行する際のプロセスと方法を検討します。たとえば、最初のパフォーマンステストまたは最初のワークフローの概念実証 (POC) で必要なデータを検討します。

データは重力です。EDA ツールを実行するのに必要な限られたデータ量のみを Amazon Simple Storage Service (Amazon S3) バケットに移行すれば、AWS でアーキテクチャを柔軟かつ迅速に構築、反復処理できます。Amazon S3 にデータを格納することから得られるメリットはいくつかあります。EC2 インスタンスへの S3 の転送速度は最大 25 Gbps なので、POC EDA では Amazon S3 を使用すると反復処理が迅速化されます。S3 バケットに格納されたデータをさまざまな EC2 インスタンスタイプですばやく実験できます。さらに、一時的な共有ファイルシステムの作成と調整など、さまざまな作業ストレージオプションを実験することもできます。

転送するデータは、POC に使用する予定のツールや設計によって異なります。1 つのシミュレーションジョブを実行するのに必要なデータのみを使用するなど、比較的少ない POC データ量から開始することが推奨されます。こうすることで、EDA POC の初期ワークロードを実行しながら AWS に短期間で慣れることができ、本番稼動用のアーキテクチャを AWS に構築する方法を理解できるようになります。

### 前提条件

半導体の設計環境には、ワークフローを AWS に移行するプロセスの妨げとなる複雑な依存関係が頻繁に発生します。最初の POC (概念実証) や複雑なアーキテクチャを構築する場合でも AWS はお客様をサポートしますが、レガシーのオンプレミスデータの依存関係は、設計者またはツールエンジニアが解明しなければなりません。最初の POC プロセスで、プロジェクトデータと一緒に移行すべき共有ライブラリなどの依存関係を特定する必要があります。依存関係のリストはツールを使用して作成できます。ツールによっては、AWS へのデータ移行プロセスを効率化するファイルマニフェストを生成できます。たとえば、Ellexus Container Checker というツールは AWS Marketplace で入手できます。

依存関係には、認証方法 (NIS など)、共有ファイルシステム、組織の枠を超えたコラボレーション、グローバルに分散される設計などが含まれる可能性があります。(このような依存関係の特定と管理はクラウド移行に限ったものではありません。あらゆる EDA の分散環境において半導体設計チームも同じような課題に直面します)。

また、新しい半導体プロジェクトを AWS で起動してレガシーの依存関係の件数を大幅に低減する方法もあります。

### 最初の概念実証 (POC) に推奨される EDA ツール

HDL コンパイルとシミュレーションワークフローは、AWS での EDA POC の起動や、本番稼動用 EDA 環境の作成を最速で実行できるアプローチである可能性があります。

HDL ファイルは一般的に大きくありません。また、オンプレミスのライセンスサーバー (VPN 経由) を使用する機能があるため、ライセンス環境を AWS に移行する追加の作業が不要になります。HDL コンパイルとシミュレーションワークフローは、共有ファイルシステムやジョブスケジューリングの必要性も含め、その他の EDA ワークロードを代表するものです。

## クラウド向けに最適化された従来のアーキテクチャ

AWS では、コンピューティングリソースとストレージリソースがオンデマンドで使用できるため、必要なときに必要な分だけを起動できます。これにより、異なるアーキテクチャの設計方法を半導体設計環境に使用できます。1 つの大きなクラスターで複数のプロジェクトを実行するのではなく、AWS を使用して複数のクラスターを起動できます。コンピューティングリソースをオンデマンドで増減されるように構成できるため、ワークフローのさまざまな部分に固有のクラスターや、プロジェクトに固有のクラスターを構築できます。これにより、プロジェクトベースのコスト配分、コンピューティングとストレージの適切なサイズ調整、環境分離といったさまざまなメリットが得られます。

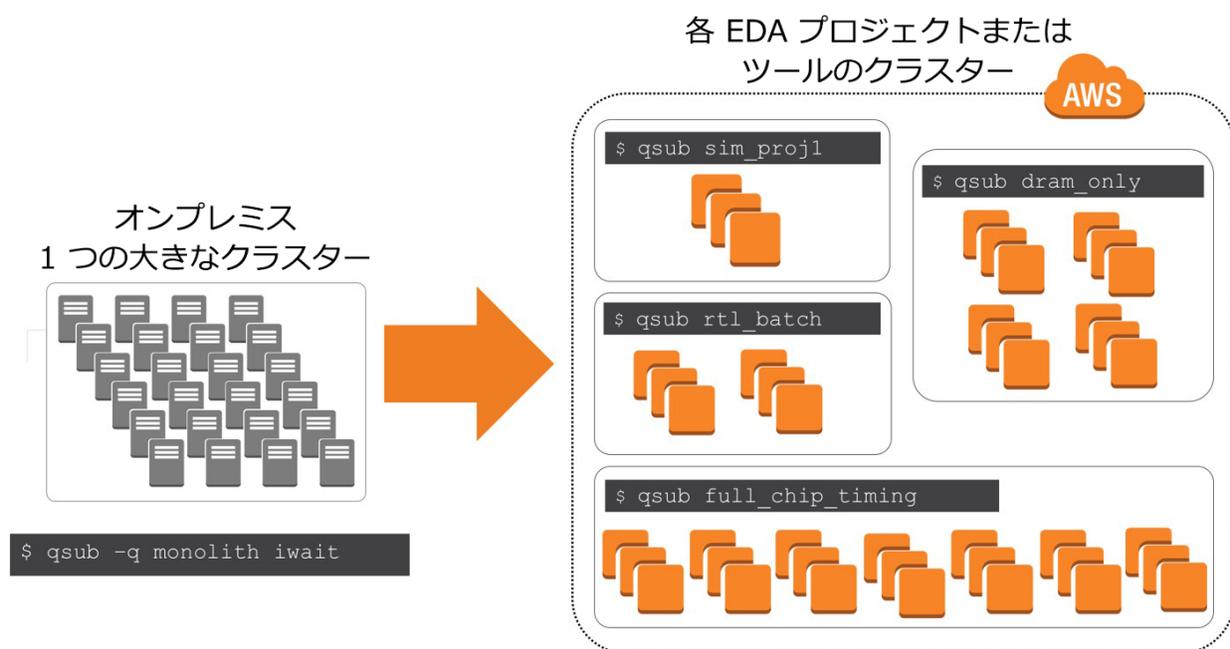


図 1: AWS でのワークロード固有の EDA クラスター

図 1 に示すように、AWS に移行すると、EDA ワークロード (クラスターなど) ごとに個別のリソースセットを起動できるようになります。このマルチクラスターを利用するアプローチは、グローバルな組織の枠を超えたコラボレーションでも使用できます。特定のプロジェクトに対して固有のクラウドリソースを専有して管理するなど、マルチクラスターを利用するアプローチによって、組織はプロジェクトに必要なリソースのみを使用できるようになります。

## ジョブスケジューラの統合

AWS で構築する EDA ワークフローは、オンプレミスのデータセンターで使用する環境に似ています。多くの場合、データセンターで実行している EDA ツールとアプリケーションやオーケストレーションソフトウェアは AWS でも実行できます。IBM Platform LSF、Adaptive PBS Pro、Univa Grid Engine (またはオープンソースの代替ツール) などのジョブスケジューラは、コンピューティングリソースの管理、ライセンス使用量の最適化、ジョブの調整と優先順位付けに EDA 業界で一般的に使用されます。AWS に移行する際、これらの既存のスケジューラを基本的には変更せずに、エンドユーザーのワークフローとプロセスへの影響を最小限に抑えることもできます。ほとんどのジョブスケジューラでは AWS と何らかのネイティブ統合がすでに行われているため、保留中のジョブがキュー内にある場合はマスターノードを使用して自動的にクラウドリソースを起動できます。AWS でリソースの割り当てと管理を自動化する手順については、特定のジョブ管理ツールのドキュメントを参照してください。

## AWS での EDA アーキテクチャの構築

本番稼働用 EDA ワークフローを AWS で構築するには、現在の環境をエンドツーエンドで検証する必要があります。この検証は、EDA ツールの実行に使用するオペレーティングシステムの他、ジョブスケジューリング、ユーザー管理環境から開始します。AWS では半導体の設計ワークロードを移行する際にアーキテクチャを混在でき、次の 2 つのアプローチを組み合わせることができます。

- 従来のジョブスケジューリングソフトウェアを使用して従来のクラスターに類似するアーキテクチャを構築する。ただし、クラウドネイティブな手法を必ず使用する。
- アプリケーションのコンテナ化を使用する AWS Batch などのクラウドネイティブな方法を使用する。

必要に応じて、大規模な並列パラメータスイープを実行する場合など、AWS Batch を使用すると効率化される場合を特定します。

## ハイパーバイザー: Nitro、Xen

Amazon EC2 インスタンスではハイパーバイザーを使用してサーバーでリソースを分割するため、お客様のインスタンス専用に独立した CPU、メモリ、ストレージリソースを使用できます。ただし、T\*ファミリー以外のインスタンスではハイパーバイザーによるリソース共有は行なっていません。C4 や R4 ファミリーなどの旧世代のインスタンスタイプでは、EC2 インスタンスは Xen ハイパーバイザーを使用して仮想化されます。C5、R5、

Z1d などの現行世代のインスタンスでは、専用のハードウェアと KVM に基づいて高度にカスタマイズされたハイパーバイザーを使用します。この新しいハイパーバイザーシステムは Nitro と呼ばれます。このドキュメントの作成時点での Nitro ベースのインスタンスは、Z1d、C5、C5d、M5、M5d、R5、R5d です。

Nitro ベースのインスタンスを起動するには、ネットワーキングとストレージ用に特定のドライバをインストールし、インスタンスを起動する前に有効にしておく必要があります。この構成については、次のセクションで詳しく説明します。

## AMI およびオペレーティングシステム

AWS にはさまざまなオペレーティングシステム (OS) のサポートが組み込まれています。EDA ユーザーの場合、他のオペレーティングシステムよりも CentOS、Red Hat Enterprise Linux、Amazon Linux 2 が使用されます。オペレーティングシステムとオンプレミス環境で行ったカスタマイズは AWS で EDA アーキテクチャを構築するベースラインとなります。EC2 インスタンスを起動する前に、使用する Amazon Machine Image (AMI) を決定する必要があります。AMI には OS が含まれ、OS とドライバへの必要なカスタマイズはすべて組み込まれています。さらに、アプリケーションソフトウェアを含めることもできます。EDA の場合、既存の AMI からインスタンスを起動し、起動後にそのインスタンスをカスタマイズして、更新されたこの構成をカスタム AMI として保存するアプローチがあります。この新しいカスタム AMI から起動されるインスタンスには、AMI の作成時に組み込まれたカスタマイズが含まれます。

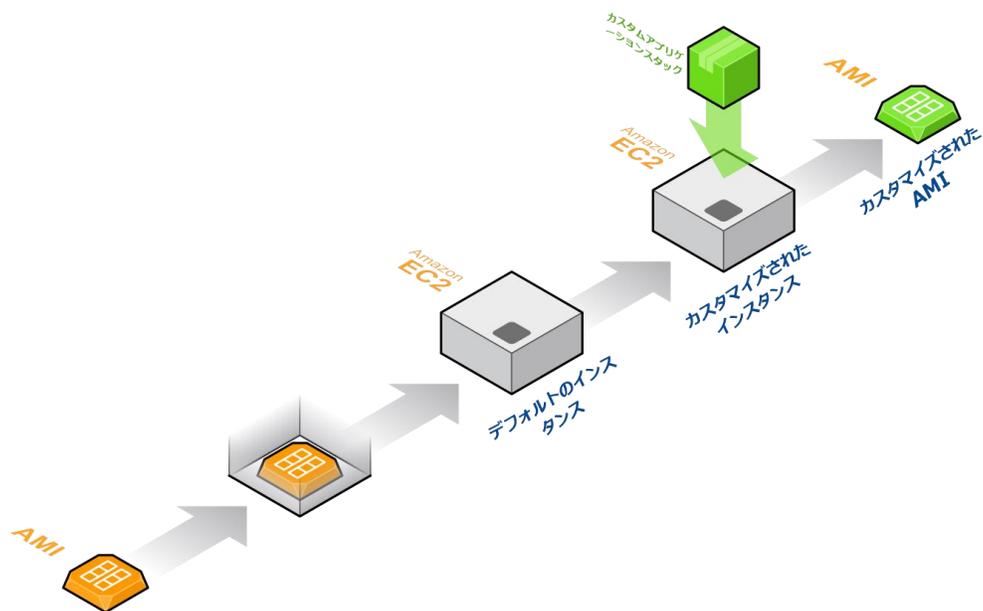


図 2: Amazon が提供する AMI を使用してカスタマイズされた AMI を構築する

図 2 は、AMI を使用してインスタンスを起動するプロセスを示しています。AMI は AWS コンソールまたは AWS Marketplace から選択できます。その後、そのインスタンスをお使いの EDA ツールと環境でカスタマイズできます。その後、カスタマイズされたインスタンスを使用して、AWS で全体の EDA 環境の起動に使用できるカスタマイズされた新しい AMI を作成できます。また、このプロセスで作成したカスタマイズされた AMI は、さらにカスタマイズすることが可能です。たとえば、AMI をカスタマイズしてアプリケーションソフトウェアをさらに追加したり、追加のライブラリのロード、パッチの適用をしたりできます。その度に EC2 インスタンスでカスタマイズされた AMI が起動します。

本文書の執筆時点では、EDA ツールとして以下に示すレベルの OS を お勧めします (OS) のバージョンの詳細については、次のセクションに記載されています)。

- Amazon Linux と Amazon Linux 2 (EDA ツールのベンダーと証明書の確認を行ってください)
- CentOS 7.4 または 7.5
- Red Hat Enterprise Linux 7.4 または 7.5

これらの OS レベルには、Nitro ベースのインスタンスを含む現在のインスタンスタイプのサポートに必要なドライバーがすでに含まれています。これらのいずれかのレベルを使用していない場合は、現在のインスタンスの機能を利用するために追加のステップを実行する必要があります。具体的には、Elastic Network Adapter (ENA) ドライバーに依存する 拡張 ネットワーキングの構築と有効化が必要です。ENA ドライバーと AMI ドライバーについて詳しくは、**Network と Optimizing EDA Tools on AWS** を参照してください。

Nitro (Z1d、C5、C5d、M5、M5d、R5、R5d) でインスタンスを使用する場合、AWS の ENA ドライバーが構築、有効化された AMI を使用し、NVMe ドライバーがインストールされている必要があります。現時点ではこれらのドライバーがない限り、Nitro ベースのインスタンスは起動しません。下記に示す OS レベルに必要なドライバーが含まれています。

- CentOS 7.4 以降
- Red Hat Enterprise Linux 7.4 以降
- Amazon Linux または Amazon Linux 2 (現在のバージョン)

Nitro ベースのインスタンスで AMI を起動できることを確認するには、まず Xen ベースのインスタンスタイプで AMI を起動し、[awslabs/aws-support-tools/EC2/C5M5 InstanceChecks/](https://aws.amazon.com/jp/support-portal/faq/instance-checks/) の AWS ラボの GitHub repo にある `c5_m5_checks_script.sh` スクリプトを実行します。

このスクリプトは、AMI を分析し、Nitro ベースのインスタンスで実行できるかどうかを確認します。実行できない場合は、推奨される変更が表示されます。

ユーザーの AMI で使用する独自のオンプレミスのイメージをインポートすることもできます。この処理には、追加の手順が必要となりますが、結果的に時間の節約ができます。オンプレミスの OS イメージをインポートする前に、まず、OS の VM イメージが必要です。AWS は、S3 バケットにアップロードした後に AMI に変換される必要がある、特定の VM 形式 (VMware ESX 上の Linux 仮想マシンなど) をサポートしています。詳細な情報と使用説明については、<https://aws.amazon.com/ec2/vm-import/>で確認できます。

。

上述のオペレーティングシステムと同様の要件が、インポートされたイメージにも適用されます (CentOS/RHEL 7.4 または 7.5、Amazon Linux または Amazon Linux 2 を使用する必要があります)。

## コンピューティング

AWS にはさまざまなインスタンスのタイプやサイズがありますが、コンピューティング最適化インスタンスのインスタンスタイプやメモリ最適化カテゴリは、通常 EDA ワークロードに最も適しています。AWS で EDA ソフトウェアを実行する場合、最新世代のインテル Xeon プロセッサの機能を持ち、それがワークフロー全体の各アプリケーションの要件に合う異なるいくつかの設定を使用するものを選択する必要があります。

コンピューティング最適化インスタンスファミリーは、AWS で利用可能な最も高いクロック周波数を持つインスタンスを特徴としており、通常いくつかの集約的なワークロードの実行に十分なメモリインスタンスを持ちます。

以下に示すのは、コンピューティング最適化されたインスタンスタイプの典型的な EDA のユースケースです。

- シミュレーション
- 合成
- フォーマル検証
- 回帰テスト

## EDA ツール用の Z1d

AWS は近頃、EDA アプリケーションに対して十分に最適化された強力な新しいインスタンスタイプを発表しました。Z1d インスタンスのより速いクロック速度と、最大 4GHz を維持するターボ実行によって、より早く結果を出しながら、EDA ライセンスの最適化が可能となります。Z1d は AWS 固有のインテル Xeon Platinum 8000 シリーズ (Skylake) プロセッサを使用する、最速の AWS インスタンスタイプです。以下に Z1d インスタンスの機能を要約します。

- すべてのコア周波数を最大 4.0 GHz に維持
- インスタンスあたり最大 24 コア (48 スレッド) の異なる 6 つのインスタンスサイズ
- 合計 384 GiB のメモリ
- コアあたり 16 GiB RAM のコアレシオのメモリ
- ローカルインスタンスストア NVMe ストレージ (1.8 TiB 程度) が含まれます。
- EDA やその他の高性能なワークロード用に最適化

## 追加のコンピューティング最適化インスタンス

### C5、C5d、C4

Z1d の他、C5 インスタンスには最大 36 個のコア (72 スレッド) と RAM 最大 144 GiB の機能があります。C5 で使用されているプロセッサは Z1d と同じインテル Xeon Platinum 8000 シリーズ (Skylake) ですが、さらに 3.0 GHz のベースクロック速度と最大 3.5 GHz までターボブーストできます。

C5d インスタンスは C5 と同じ設定ですが、1.8 TiB ものローカル NVMe SSD ストレージを提供します。

旧世代の C4 インスタンスも EDA のお客様に一般的に利用されており、メモリ負荷の高くない特定のワークロードなどに対しては引き続き適切なインスタンスです。

## メモリ最適化インスタンス

### Z1d、R5、R5d、R4、X1、X1e

Z1d インスタンスではコンピューティング最適化のみでなく、合計メモリ 384 GiB を含むメモリ最適化が施されています。Z1d はすべてのインスタンス中、最も高いクロック周波数を持ち X1 および X1e インスタンスを除き、コアあたり最大のメモリ (16 GiB/コア) と

等しくなります。Z1d で利用可能なメモリより大きいメモリが必要な場合、R5、R5d、R4、X1、X1e などのその他のメモリ最適化されたインスタンスの利用をご検討ください。下記にメモリ最適化されたインスタンスタイプの典型的な EDA のユースケースを示します。

- 配置配線
- 静的タイミング解析
- フィジカル検証
- バッチモード RTL シミュレーション (マルチスレッドに対して最適化されたツール)

R5 および R5d は、Z1d と C5 と同じインテル Xeon Platinum 8000 シリーズ (Skylake) のプロセッサを持ちます。最も大きい R5 および R5d インスタンスタイプで最大 768 GiB のメモリを備えているため、以前は X1 または X1e でのみ EDA ワークロードを実行できましたが、現在は R5 と R5d インスタンスで実行できるようになりました。これらの最近リリースされたインスタンスは、R4 インスタンスの完全互換であるため、配置およびルート、バッチモード RTL シミュレーションに使用できます。R4.16xlarge インスタンスは、高いコア数 (32) と 15 GiB/コアを備えた、実行可能なオプションです。このため、多くのお客様が R4.16xlarge インスタンスタイプを使用しています。

X1 および X1e インスタンスタイプはメモリ負荷の高いワークロードにも使用できます。ただし Amazon の社内シリコンチームによる EDA ツールの試験では、ほとんどの EDA ツールは Z1d、R4、R5、R5d インスタンスで問題なく動作することが示されました。半導体設計では、X1 (1952 GiB) および X1e (3904 GiB) で利用可能なメモリ量が必要となることは、あまりありませんでした。

## ハイパースレッディング

Amazon EC2 インスタンスは、単一のインテル Xeon CPU コアで複数のスレッドの同時実行を可能にする、インテルハイパースレッディングテクノロジー (HT テクノロジー) をサポートしています。各スレッドはインスタンスの仮想 CPU (vCPU) を表します。インスタンスには、インスタンスタイプによって異なるデフォルトの CPU コア数があります。各 vCPU は、T2 インスタンスを除いて、インテル Xeon CPU コアのハイパースレッドです。以下の CPU オプションを指定して、インスタンスを半導体設計ワークロード用に最適化できます。

- **CPU コアの数:** インスタンスの CPU コア数をカスタマイズできます。このカスタマイズによって、メモリ負荷の高いワークロードのための RAM を十分に持ちながらも、少ない CPU コアのインスタンスを用いて、ソフトウェアのライセンス費用を最適化できます。

- コアごとのスレッド:** 各 CPU コアに単一のスレッドを指定することで、インテルハイパースレッディングテクノロジーを無効にできます。このシナリオはハイパフォーマンスコンピューティング (HPC) のワークロードのような、特定のワークロードに当てはまります。

インスタンスの起動時にこれらの CPU オプションを指定できます。(現在のところ AWS コマンドラインインターフェース [AWS CLI]、AWS ソフトウェア開発キット [SDK]、Amazon EC2 API のみでサポートされています)。CPU のオプションの指定に料金の追加や減額は発生しません。デフォルトの CPU オプションで起動されたインスタンスと同じ料金が発生します。CPU オプション指定の詳細とルールについては、**Amazon Elastic Compute Cloud - Linux インスタンス用ユーザーガイド** の[CPU オプションの最適化](#)を参照してください。

vCPU の数を 2 で割り、インスタンスの物理的なコア数を出します。アプリケーションのパフォーマンスに悪影響があると判断した場合、HT テクノロジーを無効にできます。ハイパースレッディングの無効化についての詳細は、**Optimizing EDA Tools on AWS** を参照してください。

表 1 に、EDA ツールに使用される一般的なインスタンスタイプを一覧にしています。

表 1: EDA ワークロードに適したインスタンス仕様

インスタンス名	*最大物理コア数	CPU クロック周波数	最大メモリ (GiB)	コアあたりのメモリ GiB/コア	ローカル NVMe
Z1d	24	4.0 GHz	384	16	はい
R5/R5d	48	最大 3.1 GHz	768	16	はい (R5d の場合)
R4	32	2.3 GHz	488	15.25	
M5/M5d	48	最大 3.1 GHz	384	8	はい (M5d の場合)
C5/C5d	36	最大 3.5 GHz	144	4	はい (C5d の場合)

インスタンス名	*最大コアカウント	CPU クロック周波数	GiB あたりの最大合計RAM	コアあたりのメモリ GiB/コア	ローカル NVMe
X1	64	2.3 GHz	1,952	30.5	はい
X1e	64	2.3 GHz	3,904	61	はい
C4	18	2.9 から 3.5 GHz に増加	60	3.33	

\*注意: AWS は vCPU (インテルハイパースレッド) をプロセッサに用いています。この表ではコアを使用しています。

## ネットワーク

Amazon の拡張ネットワークテクノロジーでは、現行世代のインスタンスは最大 25 Gbps、旧世代のインスタンスは最大 10 Gbps での通信が可能です。さらに、拡張ネットワークによってレイテンシーとネットワークのストレスが減少しました。

下記に示すオペレーティングシステムのレベルで、拡張ネットワークがデフォルトで有効にされています。

- Amazon Linux
- Amazon Linux 2
- CentOS 7.4 および 7.5
- Red Hat Enterprise Linux 7.4 および 7.5

CentOS または RHEL の旧バージョンをご利用の場合は、ネットワークモジュールをインストールして、インスタンス用の拡張ネットワークアダプター (ENA) サポートの属性を更新することで、拡張ネットワークを有効にできます。設定とインストール方法を含む、拡張ネットワークの詳細については、**Amazon Elastic Compute Cloud - Linux インスタンス**用ユーザーガイドの[Linux での拡張ネットワーク](#)ページを参照してください。

## ストレージ

任意のインフラストラクチャで、そのスケールで実行されている EDA ワークロードは、キューでジョブをプッシュするために、ストレージがすぐにボトルネックとなってしまいます。ネットワークファイルシステム (NFS) で使われる従来の一元化されたファイラーは、一般的に高い EDA スループットのサポートのために高額な費用でハードウェアのベンダーから購入されています。しかしながら、これらの一元化されたファイラーはすぐに EDA でボトルネックとなるため、ジョブの実行時間が増加すると同時に EDA ライセンス料が高くなってしまいます。計画的または予想外の EDA データの増加や、急増する EDA クラスタ内でのデータアクセスの必要性によって、ファイラーは結果的にストレージ空間を使い果たすか、ネットワークまたはストレージ層によって帯域幅が制限されることとなります。

EDA アプリケーションは、AWS で利用可能な幅広いストレージのオプションを活用できるため、大規模なバッチ処理の実行時間を結果的に削減できます。これらの利点を実現するにはいくらかの EDA ワークフローの再設定が必要となることがありますが、最適化を行う利点は大きいといえます。

## AWS のストレージのタイプ

EDA ストレージをデプロイする様々なオプションについて説明する前に、AWS で利用可能なストレージサービスの種類について理解することが重要です。

### Amazon EBS

Amazon Elastic Block Store (Amazon EBS) は、AWS クラウドの Amazon EC2 インスタンスで使用する永続的なブロックレベルのストレージボリュームを提供します。EBS ボリュームは広帯域のネットワークファブリック上のインスタンスに付属しており、インスタンス自体のファイルシステムと設定可能なローカルのブロックストレージとして表示されます。コンポーネントに障害が発生した場合でも高い可用性と耐久性を提供できるように、各 Amazon EBS ボリュームはアベイラビリティゾーン内で自動的にレプリケートされます。Amazon EBS ボリュームは、半導体のワークロードの実行に必要な一貫性を持つ、低レイテンシーのパフォーマンスを提供します。

インスタンスタイプを選択する際、デフォルトで Amazon EBS に最適化されたインスタンスを選ぶ必要があります。Amazon EBS に最適化されたインスタンスは、その他すべてのネットワークトラフィックから隔離された Amazon EBS 専用のスループットおよび最適な Amazon I/O パフォーマンスを提供するように構成されたスタックを提供します。Amazon EBS 用に最適化されていないインスタンスを選択した場合、`-ebs-optimized` を AWS CLI の `modify-instance-attribute` パラメータと共に使用して、Amazon EBS の最適化を有効にできますが、追加料金が発生する場合があります (デフォルトで Amazon EBS が最適化されている場合、費用はインスタンスに含まれています)。

Amazon EBS は、すべての最新の Amazon EC2 インスタンス (いくつか例外があります) に対応するストレージで、AWS のファイルシステムを高速化する基礎となります。Amazon EBS を使用すると、1 つの Amazon EC2 インスタンスで最大 80,000 IOPS、1,750 MB/ 秒というパフォーマンスの達成が可能です。

AWS で EDA アーキテクチャの構築を行う際は、正しい EBS ボリュームタイプを選択することが重要です。表 2 に考慮すべき EBS ボリュームタイプを示します。

表 2: EBS ボリュームタイプ

	io1	gp2*	st1	sc1
<b>ボリュームタイプ</b>	プロビジョンド IOPS SSD	汎用 SSD	スループット最適化 HDD	コールド HDD
<b>ボリュームサイズ</b>	4 GB~16 TB	1 GB~16 TB	500 GB~16 TB	500 GB~16 TB
<b>最大 IOPS**/ ボリューム</b>	32,000	10,000	500	250
<b>最大スループット/ ボリューム</b>	500 MB/秒	160 MB/秒	500 MB/秒	250 MB/秒

\* デフォルトのボリュームタイプ

\*\*16K I/O サイズに基づく io1/gp2、1 MB I/O サイズに基づく st1/sc1

EBS ボリュームタイプを選択するときは、各 EBS ボリュームのパフォーマンス特性を考慮してください。これは NFS サーバーまたは別のファイルシステムソリューションを構築する際に特に重要です。EBS ボリュームの最大のパフォーマンスを達成できるかは、ボリュームのサイズによって異なります。さらに、gp2、st1、sc1 ボリュームタイプがバーストクレジットシステムを使用していることも考慮する必要があります。

各 AWS EC2 インスタンスタイプには、スループットと IOPS 制限があります。たとえば、Z1d.12xlarge には 1.75 GB/秒および 80,000 IOPS の EBS 制限があります。(各インスタンスタイプで予想される Amazon EBS のスループットを示すチャートについては、[Amazon Elastic Compute Cloud - Linux インスタンス用ユーザーガイドの EBS 最適化をサポートするインスタンスタイプ](#) を参照してください)。これらの速度を達成するには、各ボリュームに個別のスループットと IOPS 制限があることから、複数の EBS ボリュームを共にストライプ化する必要があります。スループット、IOPS、バーストクレジットについて詳しくは、[Amazon Elastic Compute Cloud - Linux インスタンス用ユーザーガイドの Amazon EBS ボリュームの種類](#) を参照してください。

### 動的な EBS ボリュームによるスケーラビリティの強化

半導体設計では長い間、顧客の仕様を受け取ってから数カ月または数年の間は動作しない可能性のあるバックエンドワークロードに対応するために、過剰にハードウェアをプロビジョンしてきました。AWS では、必要な時に必要なリソースのみをプロビジョンします。一般的なオンプレミスの EDA クラスタでは、初期の必要量が比較的少ない場合でも、IT チームは習慣的に大規模なネットワーク接続ストレージ (NAS) を購入しています。

EBS ストレージの重要な機能は、伸縮自在なボリューム (現行世代の EC2 インスタンスに付属するすべての EBS で利用可能) です。この機能により、現在のアプリケーションに必要なボリュームをプロビジョンして、必要量が変わった際にボリュームサイズの増量、パフォーマンスの調整、またはボリュームの使用中にボリュームの種類を変更できます。変更が反映されている間も、アプリケーションは引き続き使用できます。

オンプレミスでのインストールでは通常、ストレージ設定の調整に手動の操作を必要とします。EBS の伸縮自在なボリュームと AWS のサービスを活用することで、EBS ボリュームのサイズ変更処理を自動化できます。図 3 は Amazon CloudWatch (メトリクスおよびモニタリングサービス) と AWS Lambda (インベント駆動型、サーバーレスのコンピューティングサービス) を使用したボリューム増量の自動処理を示しています。ボリューム増量のイベントは、CloudWatch アラームおよび Lambda 関数を使用して (使用量のしきい値などを基準に) トリガーされます。結果として得られる増量はオペレーティングシステムによって自動的に検出され、続くファイルシステムの増加オペレーションでファイルシステムのサイズが変更されます。

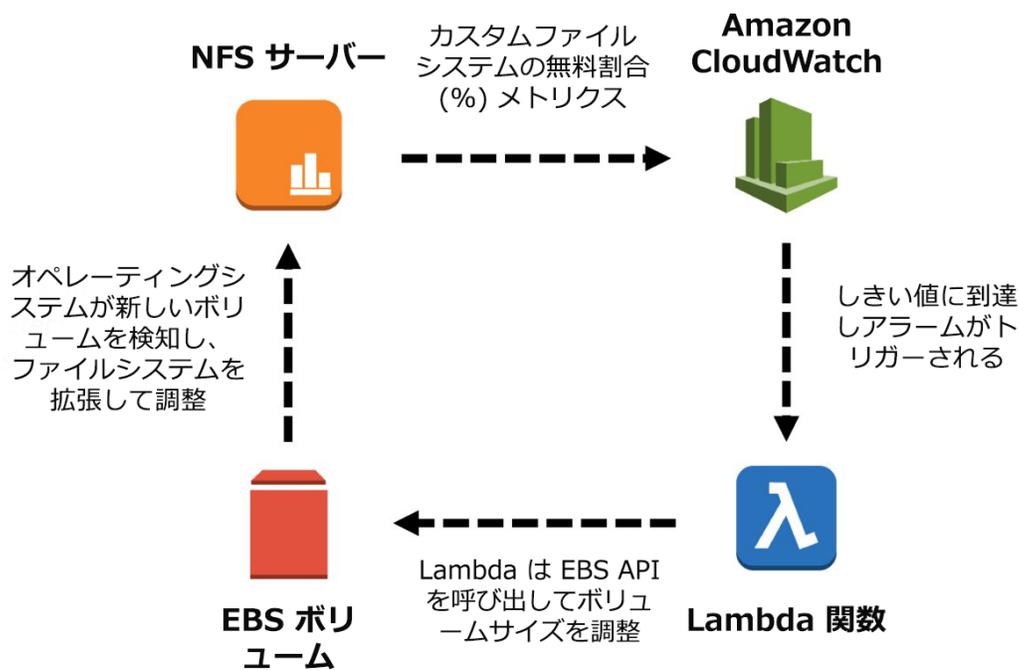


図 3: EBS ボリュームの自動的なサイズ変更のライフサイクル

## インスタンスストレージ

Amazon EBS のパフォーマンスが単一のインスタンスで十分ではない場合、インスタンスストア付きの Amazon EC2 インスタンスが利用できます。インスタンスストアは、インスタンスに物理的に付属されているブロックレベルのストレージです。ストレージが直接インスタンスに付属しているため、Amazon EBS と同様に、ネットワークベースのストレージ経由で利用可能なものよりも著しく高いスループットと IOPS の提供が可能となります。ただし、ストレージはインスタンスにローカルに付属されているため、インスタンスを停止または終了するとインスタンスストアのデータは継続されません。

さらに、インスタンスのハードウェア障害によってデータの損失という結果を招くおそれがあります。このような理由から、インスタンスストアは一時的なスクラッチスペースとして、またはインスタンスから隔離した場所 (Amazon S3 など) でデータをレプリケートすることが推奨されます。複数の NVMe デバイス付きのインスタンスを選択し、1 つまたは複数の同等のデバイスで RAID を作成することで耐久性を高めることができます。

I3 インスタンスファミリーおよび最近発表された Z1d、C5d、M5d、R5d インスタンスは、スクラッチデータなど著しい量の高速ローカルストレージを必要とする EDA ワークロードに適合しています。これらのインスタンスは NVMe ベースのストレージデバイスを使用し、また可能な限り高い IOPS を実現するように設計されています。Z1d および C5d インスタンスはそれぞれ最大 1.8 TiB のローカルインスタンスストアを備え、R5d および M5d インスタンスはそれぞれ最大 3.6 TiB のローカルインスタンスストアを備えています。

i3.16xlarge は、330 万のランダム IOPS を 4 KB ブロックサイズで実現し、最大 16 GB/秒のシーケンシャルディスクスループットを実現します。このパフォーマンスにより、i3.16xlarge は NFS 上のスクラッチまたは一時的なデータ用のファイルシステムを取り扱うのに最適です。

表 3 には、半導体分野において典型的に見られる、インスタンスストアを持つインスタンスタイプを示しています。

表 3: EDA 分野において典型的に見られる、インスタンスストアを持つインスタンス

インスタンス名	最大 Raw サイズ TiB	NVMe SSD の数とサイズ (GiB)
I3	15.2 TiB	8 x 1920
Z1d	1.8 TiB	2 x 900
R5d	3.6 TiB	4 x 900
M5d	3.6 TiB	4 x 900
C5d	1.8 TiB	2 x 900
X1	3.840 TiB	2 x 920
X1e	3.840 TiB	2 x 1920

NVMe インスタンスストレージのデータは、インスタンスのハードウェアモジュールに実装された XTS-AES-256 ブロック暗号を使用して暗号化されています。NVMe インスタンスストレージデバイスごとに固有の暗号化キーが、ハードウェアモジュールを使用して生成されます。すべてのインスタンスが停止または終了されると、暗号キーは破棄され、復元することはできません。この暗号化を無効にすることはできず、また独自の暗号化キーを指定することはできません<sup>1</sup>。

## EC2 インスタンスでの NVMe

Nitro Hypervisor をベースとする Amazon EC2 インスタンスは、ローカルの NVMe SSD ストレージを備えており、さらに、Amazon Elastic Block Store (Amazon EBS) のボリュームを NVMe ブロックデバイスとして公開します。そのため、Nitroベースのインスタンスには一定レベルのオペレーティングシステムが必要です。言い換えると、必要な NVMe ドライブがインストールされた AMI のみが、Nitro ベースのインスタンスを起動できるということです。Nitro ベースのインスタンスで AMI が実行されることを確認する方法については **AMI and Operating System** を参照してください。

Nitro ベースのインスタンスで EBS ボリュームを使用する場合は、最適なパフォーマンスを得るために 2 つのカーネル設定を構成します。詳細については、**Amazon Elastic Compute Cloud - Linux** インスタンス用ユーザーガイド「[NVMe EBS Volumes](#)」ページを参照してください。

## Amazon Elastic File System (Amazon EFS)

AWS に独自の NFS ファイルサーバーを構築するか (「従来の NFS ファイルシステム」セクションで説明します)、Amazon Elastic File System (Amazon EFS) を使用して共有

NFS ファイルシステムを起動するか、選択できます。Amazon EFS は、AWS クラウドで Amazon EC2 インスタンスと共に使用するための、シンプルでスケーラブルな NFS ベースのファイルストレージです。完全マネージド型でペタバイト規模のファイルシステムである Amazon EFS では、ファイルシステムをすばやく簡単に作成および構成できるシンプルなインターフェースを提供しています。Amazon EFS では、ストレージ容量は伸縮自在で、ファイルを追加/削除すると自動的に拡大/縮小するため、アプリケーションは必要なときに必要なだけストレージを利用できます。

Amazon EFS は、高可用性と耐久性を実現するよう設計されており、大規模にデプロイされたときに高スループットを実現できます。EFS ファイルシステムに保存されたデータは、複数のアベイラビリティゾーンに、冗長性のある形で保存されます。さらに、EFS ファイルシステムにはそのシステムが存在するリージョン内にあるすべてのアベイラビリティゾーンから同時にアクセスできます。ただし、すべてのアベイラビリティゾーンでファイルシステムのアクション (作成、読み込み、更新、削除) を認識する必要があるため、複数のアベイラビリティゾーンにまたがらない従来の共有ファイルシステムと比べてレイテンシーが高くなる可能性があります。このため、ワークロードを大規模にテストして、EFS がパフォーマンス要件を満たすことを確認するのが重要です。

### Amazon S3

Amazon Simple Storage Service (Amazon S3) は、シンプルなウェブサービスインターフェースを使用して、ウェブ上のどこからでも、容量に関係なくデータを保存および取得できるオブジェクトストレージです。99.999999999% の耐久性を実現するように設計され、世界中で何百万もの同時リクエストを処理したり数兆を超えるオブジェクトを生成したりできるようにスケールします。Amazon S3 製品は次のようなストレージクラスを取り揃えています。

- 頻繁にアクセスするデータの汎用ストレージ向けの **Amazon S3 スタンダード**。
- 長期間保存するが頻繁にはアクセスしないデータ向けの **Amazon S3 スタンダード - IA** (低頻度アクセス)。
- 長期的なデータアーカイブのための **Amazon Glacier**。

また Amazon S3 はオブジェクトを管理するための、設定可能なライフサイクルポリシーも提供しており、オブジェクトがライフサイクル全体にわたってコスト効率よく保存されるようにします。

通常、Amazon S3 に HTTP REST リクエスト経由でアクセスするには、AWS Software Development Kit (SDK) または AWS コマンドラインインターフェース (AWS CLI) を使用します。ls、cp、rm、sync コマンドライン操作を使用して他のリモートファイルシステムにデータをコピーするのと同じ方法で、AWS CLI を使用して Amazon S3 との間でデータ

タをコピーできます。

EDA ワークフローにおいて、データのアップロードおよびダウンロードを管理し高いデータ耐久性を実現するにあたっては、主要データストレージソリューションとして Amazon S3 を検討することをお勧めします。たとえば、Amazon S3 から Amazon EC2 インスタンスや Amazon EBS ストレージにすばやく効率的にデータをコピーして、大規模なバッチ回帰テストやタイミング分析を行う前に高パフォーマンスの共有ファイルシステムを作ることができます。ただし、パフォーマンスが重要となるアプリケーションの実行中に、Amazon S3 を使用して個々のファイルへの直接アクセス (読み取り/書き込み) は行わないことをお勧めします。AWS における高いパフォーマンスとデータ集約型のコンピューティングを実現するための最善のアーキテクチャは、Amazon S3、Amazon EC2、Amazon EBS、そして Amazon EFS で構成され、特定のアプリケーションごとに対して、パフォーマンス、耐久性、スケーラビリティ、コストのバランスを取ることができます。

## 従来の NFS ファイルシステム

EDA ワークフローの移行については、お使いのオンプレミス環境に似たシステムを構築することが、ストレージを AWS に移行するための選択肢として最も一般的です。これを行うことでアプリケーションを迅速に移行することができ、アプリケーションやワークフローを再設計する必要はありません。AWS を使用すれば、十分な帯域幅と Amazon EBS のスループットを備えた Amazon EC2 インスタンスを起動し、適切な EBS ボリュームをアタッチし、NFS を使用してコンピューティングノードにファイルシステムを共有することで、簡単にストレージサーバーを作成できます。

大規模なリグレッションテストおよび検証テストを行うにあたって EDA で必要となるような、巨大なスケールに対するストレージシステムを構築する場合、ストレージシステムがそのスループットを確実に処理できるようにするための手法は多数あります。

最大級の Amazon EC2 インスタンスは、25 Gbps のネットワーク帯域幅と、Amazon EBS に対する最大 80,000 の IOPS および 1,750 MB/秒の処理速度に対応しています。データが一時的またはスクラッチデータである場合、NVMe ボリュームを持つインスタンスを使用することでバックエンドのストレージを最適化できます。たとえば、8 NVMe のボリュームがある i3.16xlarge を使用すると、最大 16 GB/秒および 300 万 IOPS の処理能力によるローカルアクセスが可能です。こうなると、i3.16xlarge への 25 Gbps ネットワーク接続はボトルネックになりますが、バックエンドのストレージはボトルネックにはなりません。このセットアップを行うと、結果的に NFS の能力を 2.5 GB/秒とすることができます。

1つのインスタンスで実行できる以上のパフォーマンスを集約的に実現する必要がある

EDA ワークロードでは、特定のマウントポイントに割り当て得られた複数の NFS サーバーを構築できます。これは一般的に、共有スクラッチ、ツールディレクトリ、および個々のプロジェクトのためにサーバーを構築することを意味します。この方法でサーバーを構築することで、サーバー、および特定のワークロードの需要に応じて割り当てられるストレージを、適切なサイズにできます。プロジェクト終了時には、Amazon Glacier のような低コストの長期ストレージソリューションにデータをアーカイブできます。次いでストレージサーバーを削除することで、追加費用を抑えることができます。

ストレージサーバーを構築する場合、さまざまな選択肢があります。Linux ソフトウェア RAID (mdadm) は、その普遍性と安定性を理由に多く使用されています。しかし近年は Linux on ZFS が普及し、EDA 分野のお客様はそのデータ保護機能と拡張機能を求めて ZFS を使用しています。ZFS を使用すれば、EBS ボリュームのグループをまとめてプールしてボリュームのパフォーマンスを高め、ポイントインタイムのロールバックができるよう 1 時間ごとのスナップショット自動取得を設定し、耐障害性を得るために他のアベイラビリティゾーンにあるバックアップサーバーにデータをレプリケートするようなソリューションを比較的簡単に構築できます。

このドキュメントの範囲外ですが、自動化と管理の度合いがより高いソリューションが必要な場合は、AWS パートナーのストレージソリューションをご検討ください。AWS で高パフォーマンスのストレージを運用するためのソリューションを提供するパートナーとして、SoftNAS、WekaIO、NetApp などが挙げられます。

### クラウドネイティブなストレージ手法

低コストで強力なスケーリング動作を実現する Amazon S3 は EDA ワークフローに最適です。それは、従来の共有ストレージシステムの必要性を低減または排除するようにワークフローを適合できるためです。クラウド向けに最適化された EDA ワークフローでは Amazon EBS のストレージと Amazon S3 が組み合わせて使用され、非常に低いコストで極めて高いスケーラビリティを実現でき、従来のストレージシステムがボトルネックになることもありません。

このようなソリューションを活用するには、EDA 組織とそれをサポートする IT チームが、特定のプロジェクト (またはジョブデッキ) にどのデータが必要か理解し要件に沿ってデータを事前にパッケージ化するために、従来のツール、無秩序なファイルシステム、多数のシンボリックリンクの長年にわたるもつれを解く必要があるかもしれません。通常このアプローチで最初のステップとなるのは、共有ファイルシステムを必要としないフロントエンドのワークフローを構築するために、静的データ (アプリケーションバイナリやコンパイラなど) を動的に変化するデータと IP から分離することです。これは最適化されたク

クラウド移行のために重要なステップであるほか、従来のオンプレミス EDA ワークフローのスケラビリティと信頼性が高まるという利点もあります。

NFS 依存の度合いが比較的低いこのアプローチで EDA ストレージを管理することで、オペレーティングシステムイメージを静的アセットと共に定期的に更新でき、インスタンスの起動時に利用できるようになります。次に、ジョブがインスタンスに送られるときには、アプリケーションの起動前にまず動的データを Amazon S3 からローカルまたは Amazon EBS ストレージにダウンロードするよう設定できます。完了すると結果が Amazon S3 に戻ってアップロードされていき、すべてのジョブが終了したときに集約され、処理されます。コンピューティングをストレージからデカップリングするこの方法により、パフォーマンスと信頼性、特にフロントエンドの RTL バッチにおけるリグレッションに関して大きなメリットが得られます。

## ライセンス

オンプレミスと AWS 双方でのほとんどの EDA ワークロードで、アプリケーションのライセンスが必要です。技術的な観点から、AWS に移行する際は管理およびアクセス用のライセンスは変更されません。

## ライセンスサーバーへのアクセス

AWS では、起動したそれぞれの Amazon EC2 インスタンスに対し、Amazon Elastic Network Interface を使用して、複製やなりすましができない一意のホスト名とハードウェア (MAC) アドレスが与えられます。したがって、従来のライセンスサーバー技術 (Flexera など) は、修正を加えることなく AWS でネイティブに動作します。またライセンスサーバーを複製できないこと (AWS が MAC アドレスの複製を認めないことでこれを防いでいます) により、EDA ソフトウェアを安全な方法でデプロイし使用できるため EDA ソフトウェアベンダーに対する信頼感が高まります。

VPN や AWS Direct Connect といった接続オプションを利用できることで、ライセンスサーバーは Amazon EC2 インスタンスを使用して AWS で運用するか、お客様自身のデータセンターで運用することができます。AWS では、VPN または AWS Direct Connect を介してクラウドリソースとオンプレミスにあるライセンスサーバー間を接続できるため、ユーザーはどこにいても、ライセンスを分割して特定のコンピューティングリソース群に割り当てることなく、シームレスにワークロードを実行できます。

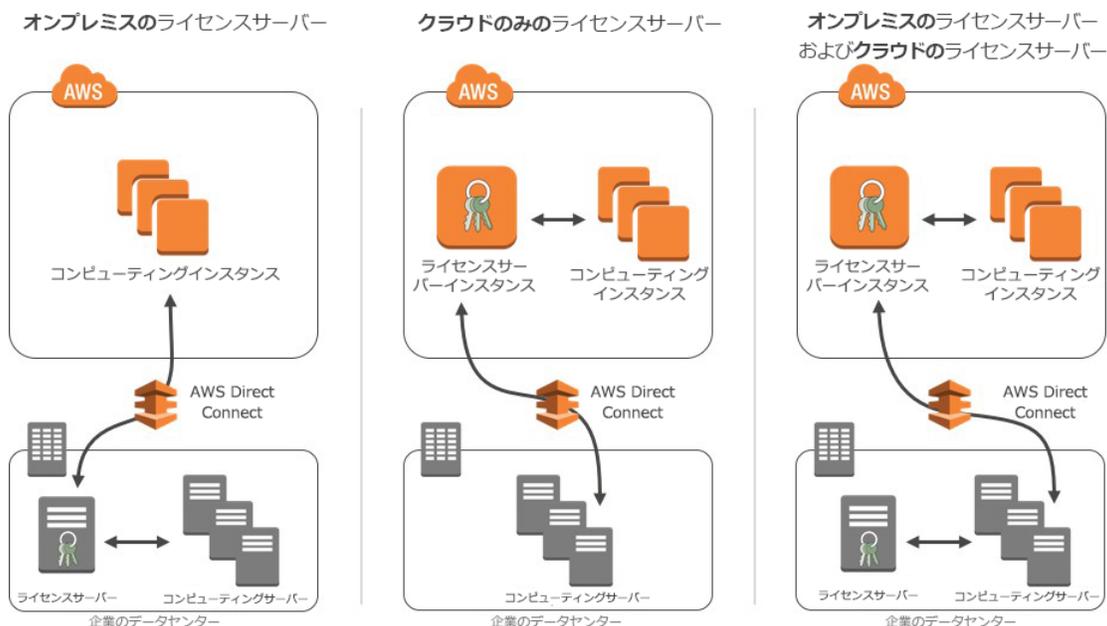


図 4: ライセンスサーバーのデプロイシナリオ

ライセンス付きのアプリケーションは、実行ホストとライセンスサーバーの間のネットワークレイテンシーやジッターの影響を受けやすい場合があります。インターネットベースの VPN は企業のデータセンターから AWS に接続するのに適した方法であることが多いものの、インターネット上のネットワークレイテンシーが変動し、ライセンス付きアプリケーションのパフォーマンスや信頼性に影響を及ぼす可能性があります。これに代わり、AWS Direct Connect を使用してオンプレミスのネットワークから最寄りの AWS リージョンに専用のプライベート接続を設けることで、レイテンシーが安定した信頼性あるネットワーク接続を実現できます。

## ライセンスサーバーの信頼性の向上

ライセンスサーバーは、ほぼあらゆる EDA コンピューティングインフラストラクチャにおける重要なコンポーネントです。ライセンスサービスが失われると、エンタープライズ全体のエンジニアリング作業が停止する可能性があります。

流動的なElastic Network Interface (ENI) を使用して AWS クラウドでライセンスをホストすることで、ライセンスサービスの信頼性を高めることができます。これらの ENI には固定されたイミュータブル MAC アドレスがあり、ソフトウェアライセンスキーと関連付けることができます。

この高可用性ソリューションを実装するには、まずライセンスサーバーインスタンスにアタッチされる ENI を作成します。お客様のライセンスキーは、このネットワークインター

フェースに関連付けられます。このインスタンスで障害が検出された場合、お客様の手で、またはカスタムのオートメーションにより、その ENI をデタッチしてスタンバイのライセンスサーバーにアタッチできます。ENI は IP アドレスと MAC アドレスを保持するため、ネットワークトラフィックは、ネットワークインターフェースが代替インスタンスにアタッチされるとすぐにスタンバイインスタンスに流れ始めます。

この独自の機能により、ライセンス管理者は、従来のデータセンターでオンプレミスサーバーを使用している場合には達成が難しかった信頼性レベルを実現できます。これは、伸縮自在でプログラム可能というクラウドの性質が持つ利点のもう一つの例です。

### EDA ベンダーとの協働

AWS は、さまざまな手法で AWS のお客様にソリューションをお届けする何千もの独立系ソフトウェアベンダー (ISV) と緊密に連携しています。こうした手法には、Software as a Service (SaaS)、Platform as a Service (PaaS)、顧客自身による管理、自分のライセンス使用 (Bring Your Own License: BYOL) などのモデルがあります。半導体部門において、AWS は EDA ソフトウェアの主要ベンダーと緊密に連携し、パフォーマンス、スケーラビリティ、コスト、アプリケーションの安全性の最適化をサポートしています。AWS は、このホワイトペーパーで説明するデプロイのベストプラクティスにより、ISV とお客様の組織を支援します。

AWS Partner Network (APN) のメンバーである EDA ベンダーは、直接提供される多様なツール、トレーニング、サポートを利用できます。これは EDA のエンドカスタマーにとっての利点です。これらのパートナープログラムの目的は、AWS が APN メンバーに提供するサポートを増やすことで個別の技術的および事業的要件に対応することです。こうしたサポートには、設計やエンジニアリング用アプリケーションを専門とする AWS パートナーチームメンバーへのアクセスなどがあります。さらに、EDA のクラウド移行において EDA ベンダーとその顧客を支援するコンサルティングパートナーの数も増えています。

### リモートデスクトップ

EDA ワークロードの大部分はバッチジョブとして実行されますが (「オーケストレーション」を参照)、EDA ユーザーがコンソールからコンピューティングサーバーに直接アクセスすることや、グラフィカルな性質のアプリケーションを使用することが必要になる場合もあります。たとえば、RTL のリグレッションエラーを特定し解決するために波形を表示したりシミュレーションを行ったりする場合や、信号の整合性の分析中に生成される結果を 2D または 3D でグラフィック表示させる場合などです。プリント回路レイアウトソフトウェアなどの一部のアプリケーションは本質的にインタラクティブであり、ユーザーにとって高品質で低レイテンシーの使用感が求められます。

そのようなアプリケーションでリモートデスクトップを AWS でデプロイする方法は複数あります。Virtual Network Computing (VNC) などのオープンソースのソフトウェア、または AWS パートナーから入手できる商用リモートデスクトップソリューションを使用できます。また、NICE Desktop Cloud Visualization (NICE DCV) や Amazon WorkSpaces などの AWS ソリューションも利用可能です。

### NICE DCV

NICE Desktop Cloud Visualization はリモート可視化テクノロジーであり、Amazon EC2 インスタンスでホストされるグラフィックを多用する 3D アプリケーションに、ユーザーが安全に接続することを可能にします。NICE DCV を使用して安全なクライアントセッションを作り出すことで、リモートユーザーが高パフォーマンスでグラフィック処理を行えるようになります。1つまたはそれ以上の EC2 インスタンスをリモートデスクトップサーバーとして使用することで、インタラクティブな EDA ユーザーは、多くのリソースを使用するアプリケーション (クラウド上で生成されたグラフィックの GPU アクセラレーションなど) を、比較的ローエンドなクライアントコンピューターでも使用できるようになります。

EDA 向けに NICE DCV を使用する一般的なシナリオでは、電磁場シミュレーションの 3D 表示や、複雑でインタラクティブな回路図キャプチャセッションのようにグラフィックを多用するアプリケーションは、ハイエンド GPU、高速 I/O 機能、そして大容量メモリを備えた高パフォーマンスの EC2 インスタンスでホストされます。

NICE DCV サーバーソフトウェアは、サーバー (EC2 インスタンス) 上でインストールおよび構成され、安全なセッションを確立するために使用されます。NICE DCV クライアントを使用してセッションにリモートで接続し、サーバーでホストされるアプリケーションを使用します。サーバーでは、ホストされているアプリケーションで要求される高い処理能力を実現するために自身のハードウェアを使用します。NICE DCV サーバーソフトウェアは、ホストされているアプリケーションの視覚的な出力を圧縮し、暗号化されたピクセルストリームとして返送します。NICE DCV クライアントは圧縮されたピクセルストリームを受信して復号し、ローカルの画面にその出力を表示します。

NICE DCV は、特に高パフォーマンスの技術的アプリケーション向けに設計されており、EDA、特に Red Hat Enterprise Linux または CentOS のオペレーティングシステムをリモートデスクトップ環境で使用している場合には絶好の選択肢です。NICE DCV は、RHEL 7 で Gnome 3 などの最新の Linux デスクトップ環境にも対応しています。

NICE DCV では、NVIDIA H.264 のハードウェアエンコーディングなどといった最新の NVIDIA Grid SDK 技術を使用しており、パフォーマンスを向上させてシステムの負荷を

軽減することができます。さらに NICE DCV では、ネットワークとプロセッサの条件が許せば可逆品質の動画圧縮もサポートされます。ネットワークの利用可能な帯域幅とレイテンシーに基づいて、動画圧縮レベルが自動的に調整されます。

## Amazon WorkSpaces

Amazon WorkSpaces はマネージド型の安全なクラウドデスクトップサービスです。Amazon WorkSpaces を使用すれば、Windows または Linux のデスクトップをわずか数分でプロビジョニングし、すばやくスケールして世界中のワーカーに数千のデスクトップを提供できます。Amazon WorkSpaces では、起動した Amazon WorkSpaces に対してのみ、月単位または時間単位でお支払いいただくため、従来のデスクトップおよびオンプレミスの仮想デスクトップインフラストラクチャ (VDI) ソリューションに比べて経済的です。Amazon WorkSpaces により、ハードウェアインベントリ、OS のバージョンとパッチ、そして VDI の管理の複雑さが解消されます。これはデスクトップ配信過程の簡素化につながります。Amazon WorkSpaces を使用すれば、エンドユーザーは、対応する任意のデバイスからいつでもどこでもアクセス可能な、高速で応答性に優れたデスクトップを選択できます。

Amazon WorkSpaces では、CPU、メモリ、およびソリッドステートストレージ (SSD) のさまざまなバンドル構成を取り揃えており、これを動的に変更できるため、アプリケーションに合わせた適切なリソースを得ることができます。必要なデスクトップ数やデスクトップのあるべき構成を予測することに時間を費やす必要がなくなるので、余計なハードウェアを購入する必要がなくなりコストを削減できます。

Amazon WorkSpaces は、デスクトップユーザーとアプリケーションを一元管理する組織にとって、またリモートデスクトップ環境に Windows または Amazon Linux 2 を利用できるユーザーにとって、最高の選択肢です。

## ユーザー認証

ユーザー認証の詳細については、「**AWS クラウドでのセキュリティおよびガバナンス**」のセクションで説明します。AWS では、オンプレミス認証サーバーへの接続、AWS へのユーザー移行、またはまったく新しい認証ソリューションのアーキテクチャの設計に際して、複数の選択肢を用意しています。

## オーケストレーション

オーケストレーションとは、EDA クラスタでのコンピューティングおよびストレージリソースの動的管理を指すほか、RTL リグレッションテストや IP 解析などの複雑なワーク

フローで処理される個々のジョブの管理 (スケジューリングおよびモニタリング) も指します。これらやその他の多くの一般的な EDA ワークフローでは、処理とストレージのリソースを効率的に利用できるか否かは、EDA ソフトウェアライセンスを効率的に利用できるか否かと同様、コンピューティング環境のオーケストレーションとアーキテクチャが十分なものかどうかで決まります。

EDA ワークロード管理の柔軟性がクラウド内で新たなレベルに達し、これによりリソースとジョブのオーケストレーションがワークロードの重要な検討事項になります。AWS では、ワークロードのオーケストレーション向けに幅広いソリューションが用意されています。完全マネージド型サービスを使えば、クラスターやジョブスケジューラのプロビジョニング、設定、最適化よりもジョブの要件と結果により多くの注意を集中できます。一方、自己管理型のソリューションを使えば、クラウドネイティブのクラスターの設定とメンテナンスを自分で行い、従来のジョブスケジューラを AWS やハイブリッドシナリオで活用できます。

このドキュメントは、EDA のオーケストレーションについての可能な方法すべてを説明するものではありません。ただし、一般的なレガシー EDA 環境で使用されているのと同じオーケストレーション方法とジョブスケジューリングが、AWS においても使用できることを知っていただくことは重要です。たとえば、市販およびオープンソースのジョブスケジューリングソフトウェアは AWS に移行することができ、Auto Scaling (要求などのトリガーにตอบสนองして EDA クラスターを動的にサイズ変更するためのもの)、CloudWatch (CPU 使用量やサーバーの健全性などのコンピューティング環境をモニタリングするためのもの) およびその他の AWS のサービスを追加することで機能強化して、コストを削減しながら性能とセキュリティを向上させることができます。

### CfnCluster

CfnCluster (クラウド形成クラスター (cloud formation cluster)) は、アマゾンウェブサービス (AWS) 上でハイパフォーマンスコンピューティングクラスターをデプロイし維持するフレームワークです。CfnCluster は AWS が開発したもので、概念実証 (POC) の迅速な開始および本番デプロイの両方を容易なものにします。CfnCluster では、EDA を含めクラスター化された多数のアプリケーションがサポートされており、拡張が容易なためさまざまなフレームワークをサポートすることができます。

CfnCluster は、既存のジョブスケジューリングソフトウェアと容易に統合でき、キューの深さやその他のトリガーに応じてサーバーを自動的に起動できます。さらに、CfnCluster は、共有ファイルシステム、クラスターヘッドノード、ライセンスサーバーやその他のリソースを起動することもできます。CfnCluster はオープンソースで容易に拡張可能なため、独自のワークフロー要件を満たすことができます。

## AWS Batch

AWS Batch は、EDA ジョブなどのクラウドでの大規模なコンピューティングワークロードを、リソースのプロビジョニングやスケジューラの管理に煩わされることなく、簡単に処理するための完全マネージド型サービスです。AWS Batch とのやりとりは、ウェブコンソール、AWS CLI、AWS SDK を使用して行います。AWS Batch は、超並列ワークロードを管理するための優れた選択肢です。

## EnginFrame

EnginFrame は、クラウドまたはオンプレミスでデプロイすることができる HPC ポータルです。EnginFrame は、オープンソースや市販の幅広いバッチスケジューリングシステムと統合され、ジョブの送信、制御、データ管理を行うためのワンスストップショップです。

AWS の EDA ユーザーは、パートナーが提供するソリューションと同様、上記のオプション (CfnCluster、AWS Batch、EnginFrame) の導入にも成功しています。特定のオーケストレーションニーズについては、AWS の技術スペシャリストにお問い合わせください。

# AWS の EDA ツールを最適化する

EDA ソフトウェアツールは、最新の半導体設計と検証にとって不可欠なものです。EDA ソフトウェアの性能の向上—個々のジョブの実行時間と EDA ジョブ全体を完了する時間の両方の関数として評価したもの—は、結果出力時間とテープアウト時間を削減し、EDA ライセンス費用を最適なものにするために重要です。

これまでの章では、AWS でのアーキテクチャ向けソリューションコンポーネントを説明してきました。ここでは、より規範的なものとなるよう、EDA ツールに期待される性能を実現するのに役立つ、特定の推奨事項と設定パラメータをご紹介します。

Amazon EC2 インスタンスタイプを正しく選択し、OS レベルを適切に最適化することは、EDA ツールを期待どおりに動作させる上で不可欠です。このセクションでは、AWS の EDA ソフトウェアツールについて実際に計測した毎日の使用量 (AWS カスタマーと Amazon 内のシリコン設計チームによる使用量) に基づいて、一連の推奨事項を示します。推奨事項には、インスタンスのタイプと設定、および、EDA ツールの代表的なセット向けの OS 推奨事項やその他のチューニング設定などの要素が含まれています。これらの推奨事項は、AWS 内部および EDA カスタマーとベンダーのテストと確認を受けたものです。

## Amazon EC2 インスタンスタイプ

次の表は、EDAツールをハイライトし、対応する推奨Amazon EC2 インスタンスタイプを示したものです。

表 4: EDA ツールと対応するインスタンスタイプ

インスタンス	*最大コア数	CPU クロック周波数	GiB あたりの最大合計 RAM と (GiB/core)	ローカル NVMe	一般的な EDA アプリケーション
<b>Z1d</b>	24	4.0 GHz	384 (16)	Y	形式的検証 RTL Simulation Batch RTL Simulation Interactive RTL Gate Level Simulation
<b>R5 / R5d</b>	48	最大 3.1 GHz	768 (16)	Y (R5d)	RTL Simulation Multi-Threaded
<b>R4</b>	32	2.3 GHz	488 (15.25)		RTL Simulation Multi-Threaded Place&Route
<b>M5 / M5d</b>	48	最大 3.1 GHz	384 (16)	Y (M5d)	Remote Desktop Sessions
<b>C5 / C5d</b>	36	最大 3.5 GHz	144 (4)	Y (C5d)	RTL Simulation Interactive RTL Gate Level Simulation
<b>X1</b>	64	2.3 GHz	1,95 (30.5) 2	Y	Place & Route Static Timing Analysis
<b>X1e</b>	64	2.3 GHz	3,90 (61) 4	Y	Place & Route Static Timing Analysis
<b>C4</b>	18	2.9 GHz (3.5 GHz にブースト)	60 (3.33)		形式的検証 RTL Simulation Interactive

\*注: AWS は vCPU (インテルのハイパースレッド) をプロセッサに用いています。この表ではコアを使用しています。

## オペレーティングシステムを最適化する

EDA ツールのインスタンスタイプを選択した後、OS をカスタマイズして最適化し、パフォーマンスを最大化する必要があります。

### 現行世代のオペレーティングシステムを使用する

Nitro ベースのインスタンスを実行している場合は、特定のオペレーティングシステムレベルを使用する必要があります。そうではなく、Xen ベースのインスタンスを実行している場合は、EDA ワークロード (特に ENA および NVMe ドライバー) には次の OS レベルの1つを引き続き使用する必要があります。

- Amazon Linux または Amazon Linux 2
- CentOS 7.4 または 7.5
- Red Hat Enterprise Linux 7.4 または 7.5

### ハイパースレッディングを無効にする

現行世代の Amazon EC2 インスタンスファミリー (T2 インスタンスファミリーを除く) では、インテル ハイパースレッディングテクノロジー (HT テクノロジー) がデフォルトで有効になっています。アプリケーションのパフォーマンスに悪影響が生じると判断される場合、HT テクノロジーは無効化できます。

次のコマンドを実行すると、各コア (物理コアおよびハイパースレッド) の詳細情報を取得できます。

```
$ cat /proc/cpuinfo
```

コアおよび対応するオンラインハイパースレッドを表示するには、`lscpu -extended` コマンドを使用します。たとえば、4 コアからなり、合計で 8 つのハイパースレッドを持つ `Z1d.2xlarge` を考えてみます。ハイパースレッディングを無効化する前後で `lscpu -extended` コマンドを実行すると、どのスレッドがオンラインでどのスレッドがオフラインかがわかります。

```
$ lscpu --extended
```

CPU	NODE	SOCKET	CORE	L1d:L1i:L2:L3	ONLINE
0	0	0	0	0:0:0:0	yes
1	0	0	1	1:1:1:0	yes
2	0	0	2	2:2:2:0	yes
3	0	0	3	3:3:3:0	yes
4	0	0	0	0:0:0:0	yes
5	0	0	1	1:1:1:0	yes
6	0	0	2	2:2:2:0	yes
7	0	0	3	3:3:3:0	yes

```
$ ./disable_ht.sh
```

CPU	NODE	SOCKET	CORE	L1d:L1i:L2:L3	ONLINE
0	0	0	0	0:0:0:0	yes
1	0	0	1	1:1:1:0	yes
2	0	0	2	2:2:2:0	yes
3	0	0	3	3:3:3:0	yes
4	-	-	-	:::	no
5	-	-	-	:::	no
6	-	-	-	:::	no
7	-	-	-	:::	no

各コアの vCPU ペア (ハイパースレッド) を表示するもう 1 つの方法は、各コアの `thread_siblings_list` を表示することです。この表には、各コアのハイパースレッドを示す 2 つの数字が表示されます。すべてのスレッドの子孫を表示するには、次のコマンドを使用するか、“\*” を CPU 番号で置き換えます。

```
$ cat/sys/devices/system/cpu/cpu*/topology/thread_siblings_list | sort -un  
0,4  
1,5  
2,6  
3,7
```

### AWS 機能 - CPU オプションを使用して HT を無効にする

CPU オプションを使用してハイパースレッディングを無効にするには、AWS CLI を `run-instances` で使用し、`-cpu-options` フラグを付加します。Z1d.12xlarge での使用例を次に示します。

```
$ aws ec2 run-instances --image-id ami-asdfasdfasdfasdf ¥  
  --instance-type z1d.12xlarge --cpu-options ¥  
  "CoreCount=24,ThreadsPerCore=1" --key-name My_Key_Name
```

CpuOptions が設定されていることを確認するには、`describe-instances` を使用します。

```
$ aws ec2 describe-instances --instance-ids i-1234qwer1234qwer  
...  
"CpuOptions": {  
  "CoreCount": 24,  
  "ThreadsPerCore": 1  
},  
...
```

### 実行中のシステムの HT を無効にする

Linux インスタンスで次のスクリプトを実行して、システムの実行中に HT テクノロジーを無効にできます。これを設定すると、`init` スクリプトで実行され、任意のインスタンスの起動時にそれが適用されます。次の例を参照してください。

```
for cpunum in $(cat/sys/devices/system/cpu/cpu*/topology/thread_siblings_list | \
  ¥ sort -un | cut -s -d, -f2-)
do
    echo 0 | sudo tee /sys/devices/system/cpu/cpu${cpunum}/online
done
```

## ブートファイルを使用して HT を無効にする

Linux カーネルが最初のセットのスレッドのみを初期化するように設定することで、HT テクノロジーを無効にすることもできます。そのためには、GRUB の maxcpus をインスタンスの vCPU 数の半数に設定します。

たとえば、Z1d.12xlarge のインスタンスの maxcpus 値を 24 とすれば、ハイパースレッディングを無効にできます。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295 maxcpus=24
```

カーネルコマンドラインの更新の方法については、「[付録 C - Linux カーネルコマンドラインの更新](#)」を参照してください。

HT テクノロジーを無効にしても、サーバーあたりのワークロード密度は変わりません。これらのツールは DRAM 容量を必要とするもので、スレッド数を減らすことは、コアあたりの GB 数が増加する場合のみ有用であるためです。

## Clocksource を TSC に変更する

Xen ハイパーバイザーを使用している旧世代のインスタンスについて、clocksource のデフォルトがハイパーバイザー内にある Xen pvclock であるため、これを TSC に更新することを検討してください。ハイパーバイザーとの通信を回避し、代わりに CPU クロックを使用するために、tsc を clocksource として使用します。

tsc clocksource は Nitro インスタンスではサポートされていません。これらのインスタンスタイプのデフォルトの kvm-clock clocksource が持つ性能上の利点は、旧世代の Xen ベースのインスタンスにある tsc と同様です。

Xen ベースのインスタンスで clocksource を変更するには、次のコマンドを実行します。

```
$ sudo su -c "echo tsc > /sys/devices/system/clocksource/clocksource0/current_clocksource"
```

clocksource が tsc に設定されていることを確認するには、次のコマンドを実行します。

```
$ cat /sys/devices/system/clocksource/clocksource0/current_clocksource  
tsc
```

インスタンスの初期化スクリプトでクロックソースを設定できます。また、次のように、dmesg コマンドを使用して clocksource が変更されたことを確認することもできます。

```
$ dmesg | grep clocksource  
...  
clocksource: Switched to clocksource tsc
```

## 深い C ステート (スリープ状態) を制限する

C ステートは非アクティブ時のコアのスリープレベルを制御します。システムのレイテンシーとパフォーマンスを調整するため、C ステートを制御することが必要な場合があります。コアをスリープ状態にするには時間がかかります。また、あるコアがスリープ状態になることで、別のコアが高い周波数で動作できる余裕が生まれますが、そのスリープ状態にあるコアが再び稼働し処理を実行するのに時間がかかります。

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0  
biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1"
```

カーネルコマンドラインの更新の方法については、「[付録 C - Linux カーネルコマンドラインの更新](#)」を参照してください。

Amazon EC2 インスタンスのプロセッサの状態についての詳細は、[Amazon Elastic Compute Cloud Linux インスタンス用ユーザーガイドの「EC2 インスタンスタイプのプロセッサのステート制御」](#)のページを参照してください。

## Xen ベースのインスタンスのターボモード (プロセッサのステート) を有効にする

現在の Nitro ベースのインスタンスタイプについては、ターボモードを変更することはできません。インスタンスごとにすでに最適な値に設定されているためです。

ソケット全体または複数のソケット (r4.16xlarge、r4.8xlarge、c4.8xlarge など) を使用している Xen ベースのインスタンスを実行している場合、特に HT テクノロジーを無効にしている場合は、ターボ周波数ブーストが活用できます。

Amazon Linux と Amazon Linux 2 ターボモードはデフォルトで有効になっていますが、その他のディストリビューションはそうになっていない場合があります。ターボモードが有効になっていることを確認するには、次のコマンドを実行します。

```
sudo su -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
```

Amazon EC2 インスタンスのプロセッサの状態についての詳細は、**Amazon Elastic Compute Cloud Linux** インスタンス用ユーザーガイドの「[EC2 インスタンスタイプのプロセッサのステート制御](#)」のページを参照してください。

## Xen ベースのインスタンスで最適な Spinlock 設定を変更する

Xen ハイパーバイザー (Nitro を除く) を使用しているインスタンスの場合、spinlock 設定を更新する必要があります。Amazon Linux、Amazon Linux 2、およびその他のディストリビューションでは、仮想マシン (VM) を低コストでプリエンプトするために最適化されたパラバーチャライズモードの spinlock が、デフォルトで実装されています。これはパフォーマンスの観点から高価となる場合があります。ロックを保持したマルチスレッドを実行する場合、VM が減速する原因となるためです。EDA ツールによっては、マルチコア用には最適化されておらず、そのため spinlock に大きく依存するものもあります。したがって、EDA ユーザーには、EC2 インスタンスのパラバーチャライズされた spinlock を無効にすることを推奨します。

Xen ベースのインスタンスでパラバーチャライズモードの spinlock を無効にするには、`/boot/grub/grub.conf` のカーネルコマンドラインに `xen_nopvspin=1` を追加し、再起動します。カーネルコマンドの例を次に示します。

```
kernel /boot/vmlinuz-4.4.41-36.55.amzn1.x86_64 root=LABEL=/  
console=tty1 console=ttyS0 selinux=0 xen_nopvspin=1
```

カーネルコマンドラインの更新の方法については、「[付録 C - Linux カーネルコマンドラインの更新](#)」を参照してください。

## ネットワーク

### AWS 拡張ネットワーク

すべてのインスタンスで必ず拡張ネットワークを使用してください。これは現在の Nitro ベースのインスタンスを起動するための要件です。ビルドとインストールの手順を含む拡張ネットワークの詳細については、**Amazon Elastic Compute Cloud Linux インスタンス用ユーザーガイド**の「[Linux の拡張ネットワーク](#)」のページを参照してください。

### クラスタープレイスメントグループ

クラスタープレイスメントグループは、単一のアベイラビリティゾーン内のインスタンスを論理的にグループ化したものです。クラスタープレイスメントグループは、ノンブロッキング、ノンオーバーサブスクリプト、フルバイセクション接続です。つまり、プレイスメントグループ内のすべてのインスタンスが、10 Gbps フローのフルラインレートでプレイスメントグループ内の他のすべてのノードと通信でき、オーバーサブスクリプションにより、減速することなく 25 Gbps のレートで集約できます。プレイスメントグループの詳細については、**Amazon Elastic Compute Cloud Linux インスタンス用ユーザーガイド**の「[プレイスメントグループ](#)」のページを参照してください。

### ネットワークの帯域幅を確認する

ENA が正しく設定されていることを確認する1つの方法は、iperf3 でインスタンス間のネットワークパフォーマンスをベンチマーキングすることです。詳細については、[Network Throughput Benchmark Linux EC2](#) を参照してください。

## ストレージ

### Amazon EBS の最適化

ワークロードのストレージ要件に合うインスタンスと EBS ボリュームを選択していることを確認します。EC2 インスタンスタイプごとに関連付けられた EBS 制限があり、各 EBS ボリュームタイプにも制限があります。たとえば、m4.16xlarge のインスタンスタイプには、スループットが最大 500 MB/ 秒の io1 ボリュームタイプがあります。

## NFS の設定と最適化

AWS に NFS サーバーを設定する前に、Amazon EC2 拡張ネットワーキングを有効にする必要があります。NFS サーバー AMI には、Amazon Linux 2 を使用することを推奨します。

高機能 NFS の重要な部分として、クライアントのマウントパラメータがあります。次に例を示します。

```
rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2
```

一般的な EFS のマウントコマンドは次の例に示すとおりです。

```
$ sudo mount -t nfs4 -o ¥  
nfsvers=4.1,rsiz=1048576,wsiz=1048576,hard,timeo=600,retrans=2 ¥  
file-system-id.efs.aws-region.amazonaws.com:/ /efs-mount-point
```

AWS で NFS サーバーを構築する場合、正しいインスタンスサイズと EBS ボリュームの数を選択します。1 つのファミリー内では通常、インスタンスが大きいほど、それが使用できるネットワークと Amazon EBS の帯域幅は大きくなります。AWS で最大の NFS サーバーは、多くの場合、m4.16xlarge インスタンスで構築され、複数の EBS ボリュームが、最大限の性能を発揮できるようにストライプ化されています。

AWS に ANF サーバーを構築するための詳細については、「[付録 A - Optimizing Storage](#)」を参照してください。

## カーネル仮想メモリ

一般的なオペレーティングシステムのディストリビューションは、AWS for EA のワークロードが提供するような大規模マシン用には調整されていません。そのため、すぐに使用できるように設定されていても、カーネルのネットワークバッファやストレージのページキャッシュのバックグラウンドドレインなどに対しては性能上最適な設定になっていません。次に指定する数値はインスタンスのサイズやアプリケーションの実行によって変動する可能性はありますが、AWS EDA チームでは、次に示すカーネルの設定値を開始点として、インスタンスのメモリ使用率を最適化することを推奨します。

```
vm.min_free_kbytes=1048576
```

```
vm.dirty_background_bytes=107374182
```

## AWS クラウドでのセキュリティおよびガバナンス

クラウドは、幅広いツールと設定を提供するため、企業はこれを利用して、従来のオンプレミスの環境では実現困難な方法で、データや IP を保護することができます。このセクションでは、AWS クラウドでデータを保護する方法の一部を紹介します。

### データ保護とデータ主権のための隔離された環境

セキュリティグループは、ファイアウォールに似ています。これらは徹底して、特定のリソースへのアクセスを厳密に管理します。コンピューティングとストレージのリソースを含むサブネットは分離されているため、インターネットに直接アクセスすることはありません。この環境にアクセスする必要のあるユーザーはまず、SSH などのセキュリティ保護されたプロトコルを介して Bastion Node (別名ジャンプボックス) に接続する必要があります。ここからは、企業のセキュリティポリシーで許可されているインタラクティブなデスクトップやジョブスケジューラにログインできます。

多くの場合、隔離された環境には、セキュリティ保護された FTP が必要です。通常、企業はセキュリティ保護された FTP を使用して、ツールを取引先からダウンロードしたり、完成した設計を製造施設にコピーしたり、サプライヤから IP を更新したりします。これを安全に行うには、必要に応じ、隔離されたサブネットに FTP クライアントを設定し、外部の IP アドレスへのアクセスに制限を加えることができます。

このクライアントを他のネットワークからセグメント化し、厳格な制御とモニタリングを設定してそのサーバーのすべてがセキュアになるよう徹底してください。

### ユーザー認証

ユーザーとコンピューティングノードへのアクセスを管理する場合、現在使用している技術を適用して、AWS 上と同様に機能させることができます。すでに多くの企業は、既存の LDAP、Microsoft Active Directory、または NIS サービスを使用して認証を行っています。これらのサービスの大半は、レプリケーションと複数のデータセンターをサポート

する機能を提供しています。適切なネットワークと VPN セットアップを用意すれば、リモートのデータセンター構成で使用するのと同じ方法と構成を使用して、AWS 上でこれらのシステムを管理できます。

企業がクラウドに隔離されたディレクトリを実行する場合、選択できるオプションは数多くあります。マネージド型ソリューションを使用する場合は、AWS [Directory Service for Microsoft Active Directory \(Standard\)](#) が広く選ばれています<sup>2</sup>。AWS Microsoft AD (Standard Edition) は、中小規模の企業 (SMB) 向けに最適化されたマネージド型の Microsoft Active Directory (AD) です。その他のオプションとして、独自の LDAP または NIS インフラストラクチャを AWS 上で実行することや、FreeIPA などの最新のソリューションがあります。

## ネットワーク

AWS には、コンポーネントを相互に分離しネットワークアクセスを制御できる、多くのテクノロジーがあります。

### Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) は、AWS クラウドの論理的に分離されたセクションのプロビジョニングを可能にします。そこでは、お客様が定義した仮想ネットワーク内で AWS リソースを起動することができます。独自の IP アドレス範囲の選択、サブネットの作成、ルートテーブルとネットワークゲートウェイの設定など、仮想ネットワーク環境を完全にコントロールできます。お客様の VPC では、リソースやアプリケーションに安全かつ簡単にアクセスできるよう、IPv4 と IPv6 の両方を使用できます。

Amazon VPC のネットワーク設定は容易にカスタマイズすることができます。たとえば、インターネットへのアクセスがある FTP と踏み台サーバーのパブリックサブネットを作成し、デザインシステムやエンジニアリングシステムをインターネットへのアクセスがないプライベートサブネットに配置できます。セキュリティグループやネットワークアクセスコントロールリストなどの複数のセキュリティレイヤーを活用し、各サブネットの EC2 インスタンスへのアクセスをコントロールすることができます。

加えて、企業のデータセンターと自分の VPC 間にハードウェア仮想プライベートネットワーク (VPN) 接続を作成することができるので、AWS クラウドを企業のデータセンターを拡張するかのよう活用することができます。

## セキュリティグループ

Amazon VPC では、セキュリティグループやネットワークアクセス制御リストなどの高度なセキュリティ機能を提供しており、インスタンスレベルとサブネットレベルでインバウンドおよびアウトバウンドでフィルタリングが可能になります。セキュリティグループは、インスタンスの仮想ファイアウォールとして機能し、インバウンドトラフィックとアウトバウンドトラフィックを制御します。VPC 内でインスタンスを起動した場合、そのインスタンスには最大 5 つのセキュリティグループを割り当てることができます。

ネットワークアクセスコントロールリスト (ACL) は、サブネットのインバウンドトラフィックとアウトバウンドトラフィックを制御します。ほとんどの場合、セキュリティグループでお客様のニーズに対応することができます。ただし、VPC に追加のセキュリティレイヤーが必要な場合は、ネットワーク ACL を使用することもできます。詳細については、Amazon Virtual Private Cloud ユーザーガイドの [セキュリティ](#) ページをご覧ください。

Amazon VPC またはサブネットでフローログを作成し、VPC またはサブネットでネットワークインターフェイスとの間で転送されるトラフィックをキャプチャできます。個別のネットワークインターフェイスでフローログを作成することもできます。フローログは Amazon CloudWatch Logs に発行されます。

## データストレージと転送

AWS では、さまざまな方法で送信中と保存中のデータを保護します。多くのサードパーティーのストレージベンダーも、AWS クラウドに実装した独自のストレージにおける特別な暗号化とセキュリティに関するテクノロジーを提供します。

## AWS Key Management Service (KMS)

AWS Key Management Service (KMS) は、データの暗号化に使用する暗号化キーの作成や制御を容易にする管理サービスです。また、Hardware Security Modules (HSM) を使用して、キーのセキュリティを保護しています。AWS Key Management Service は、Amazon EBS、Amazon S3、Amazon Redshift、Amazon Elastic Transcoder、Amazon WorkMail、Amazon Relational Database Service (Amazon RDS) を含むその他の AWS サービスと統合されており、こうしたサービスに保存したデータを保護する際に役立ちます。また AWS Key Management Service は AWS CloudTrail とも統合されており、すべてのキーの使用ログを表示できるため、規制およびコンプライアンスの要求に対応するのに役立ちます。

AWS KMS では、AWS サービスから決してエクスポートすることはできないマスターキーを作成できます。マスターキーを使用し、定義したポリシーに基づいてデータの暗号化と復号化を行います。

## Amazon EBS Encryption

Amazon Elastic Block Store (Amazon EBS) の暗号化 は、独自のキー管理インフラストラクチャの構築、管理、保護を必要とするEBS ボリューム向けの簡単な暗号化ソリューションを提供します。暗号化された EBS ボリュームを作成し、サポートされるインスタンスタイプにアタッチする場合、以下のタイプのデータが暗号化されます。

- ボリューム内の保存データ
- ボリュームおよびインスタンス間で転送されるすべてのデータ
- ボリュームから作成されたすべてのスナップショット

暗号化は EC2 インスタンスをホストするサーバーで実行され、EC2 インスタンスから Amazon EBS ストレージに転送中のデータを暗号化します。

## EC2 Instance Store Encryption

NVMe インスタンスストレージのデータは、インスタンスのハードウェアモジュールに実装されている XTS-AES-256 ブロック暗号を使用して暗号化されます。NVMe インスタンスストレージデバイスごとに固有の暗号化キーが、ハードウェアモジュールを使用して生成されます。すべての暗号化キーは、インスタンスが停止または終了されると破棄され、復元することはできません。この暗号化を無効にしたり、独自の暗号キーを指定したりすることはできません。<sup>1</sup>

## Amazon S3 Encryption

Amazon S3 の暗号化を使用すると、Amazon S3 はオブジェクトレベルでデータを暗号化します。Amazon S3では、AWS データセンターのディスクにデータが書き込まれ、データにアクセスした際にデータの復号化が行われます。要求を認証し、アクセス権を持っている限り、暗号化されたオブジェクトまたは暗号化されていないオブジェクトにアクセスする方法に違いはありません。

AWS KMS では、カスタマーマスターキー (CMK) を使用して Amazon S3 オブジェクトを暗号化します。AWS Identity and Access management (AWS IAM) コンソールの「**Encryption Keys**」セクションまたは AWS KMS API から AWS KMS を使用して、暗号化キーの作成、キーの使用法を管理するポリシーの定義、確実にキーが正しく使用され

るようにするためのキー使用の監査を実行できます。これらのキーを使用して、Amazon S3 バケットのデータを保護することができます。

AWS KMS で管理されたキー (SSE-KMS) によるサーバー側の暗号化の特徴は次のとおりです。

- ご自身で暗号化キーを作成および管理することを選択するか、カスタマー/サービス/リージョンレベルごとにデフォルトのサービスキーの作成を選択することができます。
- レスポンスの ETag はオブジェクトデータの MD5 ではありません。
- データの暗号化に使用されるデータキーも、保護対象のデータとともに暗号化されて保管されます。
- Auditable マスターキーは、IAM コンソールから作成、ローテーション、無効化することができます。
- AWS KMS のセキュリティ制御は、暗号化関連のコンプライアンス要件を満たすのに役立ちます。

Amazon S3 では、バケットに保存するすべてのオブジェクトに対してサーバー側の暗号化を必要とする場合は、受信したすべての S3 オブジェクトの復号化を行うために使用可能なバケットポリシーをサポートしています。

HTTP エンドポイント経由で Amazon S3 へのアクセスが提供されているため、バケットポリシーを活用することで、TLS 接続ですべてのデータの受信と送信が発生したことを確実にし、転送中のデータも暗号化されていることが保証されます。

## ガバナンスとモニタリング

AWSでは、AWS クラウドのデプロイメントのガバナンスとモニタリングを実施するために使用できるいくつかのサービスを提供しています。

**AWS Identity and Access Management (IAM)** – ユーザーの AWS のサービスとリソースへのアクセスを安全にコントロールできます。IAM を使用すると、AWS のユーザーとグループを作成および管理し、アクセス権を使用して AWS リソースへのアクセスを許可および拒否できます。詳細については、[AWS IAM ユーザーガイド](#)を参照してください。

**Amazon CloudWatch** – EC2 インスタンス、EBS ボリューム、S3 バケットなどの AWS リソースをほぼリアルタイムでモニタリングすることができます。CPU 使用率、

待ち時間、リクエスト数などの AWS リソースのメトリックスが、自動的に提供されます。メモリ使用率、トランザクション量、エラーレートを含む、自身のログ、カスタムアプリケーション、システムメトリックスへの CloudWatch のアクセスを提供することもでき、CloudWatch を使用してこれらをモニタリングすることもできます。詳細については、[Amazon CloudWatch ユーザー ガイド](#)を参照してください。

**Amazon CloudWatch Logs** - EC2 インスタンス、AWS CloudTrail、その他のソースからのログファイルをモニタリング、保存、アクセスするために使用します。さらに、Amazon CloudWatch Logs から関連するログデータを取得することもできます。CloudWatch にアラームを作成して、CloudTrail がキャプチャした特定の API アクティビティの通知を受け取り、通知をトラブルシューティングの実行に使用できます。詳細については、[Amazon CloudWatch Log ユーザーガイド](#)を参照してください。

**AWS CloudTrail** - AWS インフラストラクチャ全体で、API 呼び出しに関連するイベントのログの記録、継続的なモニタリング、保持が可能になります。CloudTrail では、AWS マネジメントコンソール、AWS の SDK、コマンドラインツール、その他の AWS のサービスを使用して実行される API の呼び出しなど、アカウントの AWS API の呼び出し履歴を把握できます。詳細については、[AWS CloudTrail ユーザーガイド](#)を参照してください。

**Amazon Macie** - Amazon Macie は、機械学習によって AWS 内の機密データを自動的に検出、分類、保護するセキュリティサービスです。Amazon Macie では、個人識別情報 (PII) や知的財産などの機密データが認識されます。お客様はこのサービスのダッシュボードやアラートを使用して、データのアクセスや移動の状況を確認できます。この完全マネージドサービスでは、データアクセスアクティビティの異常が継続的にモニタリングされ、不正アクセスの危険や不注意によるデータ漏洩が検出された場合には詳細なアラートが生成されます。

**AWS GuardDuty** - Amazon GuardDuty は脅威検出サービスで、悪意のある動作や不正な動作を継続的にモニタリングし、お客様が AWS のアカウントとワークロードを保護できるようにします。アカウント侵害の可能性を示す異常な API コールや不正なデプロイなどのアクティビティをモニタリングします。GuardDuty は、インスタンスへの侵入の可能性や攻撃者による偵察も検出します。

**AWS Shield** - マネージド型の分散サービス妨害攻撃 (DDoS) 防止サービスで、AWS で実行されるアプリケーションを保護します。AWS Shield ではアプリケーションのダウンタイムとレイテンシーを最小限に抑える常時稼働の検出と自動インライン緩和策を提供しているため、DDoS 保護のメリットを受けるために AWS サポートに依頼する必要はありません

。

**AWS Config** – AWS リソースの設定を評価、監査、審査するために使用します。AWS Config では、AWS リソースの設定の継続的なモニタリングと記録が行われ、適切な設定に対する記録された設定の評価を自動的に実行できます。詳細については、[AWS Config Developer Guide](#)を参照してください。

**AWS Organizations** - 複数の AWS アカウントをポリシーベースで管理できます。AWS Organizations を使用して、集中管理を行うためのサービスコントロールポリシー (SCP) を作成できます。AWS のサービスは、複数の AWS アカウントで使用します。Organizations では、一括請求 (コンソリデーティッドビルディング) を通じて組織のすべてのアカウントに対する単一の支払い方法のセットアップを可能にすることで、複数のアカウントに対する請求を簡略化します。Organizations では、アカウント内のエンティティが自社のコンプライアンスポリシー要件を満たすサービスのみ使用できるように設定できます。詳細については、[AWS Organizations ユーザーガイド](#)を参照してください。

**AWS Service Catalog** – AWS Service Catalog では、AWS での使用が承認された IT サービスのカタログを作成および管理できます。この IT サービスには、仮想マシンイメージ、サーバー、ソフトウェア、データベースから包括的な多層アプリケーションアーキテクチャまで、あらゆるものが含まれます。AWS Service Catalog では、一般的にデプロイされた IT サービスを集中管理でき、一貫性のあるガバナンスを実現し、コンプライアンス要件を満たすと同時に、ユーザーは必要な承認済みの IT サービスのみをすばやくデプロイできます。

## 寄稿者

このドキュメントの執筆に当たり、次の人物が寄稿しました。

- Mark Duffield, Worldwide Tech Leader, Semiconductors, Amazon Web Services David Pellerin, Principal Business Development for Infotech/Semiconductor, Amazon Web Services
- Matt Morris, Senior HPC Solutions Architect, Amazon Web Services
- Nafea Bshara, VP/Distinguished Engineer, Amazon Web Services

## ドキュメントの改訂

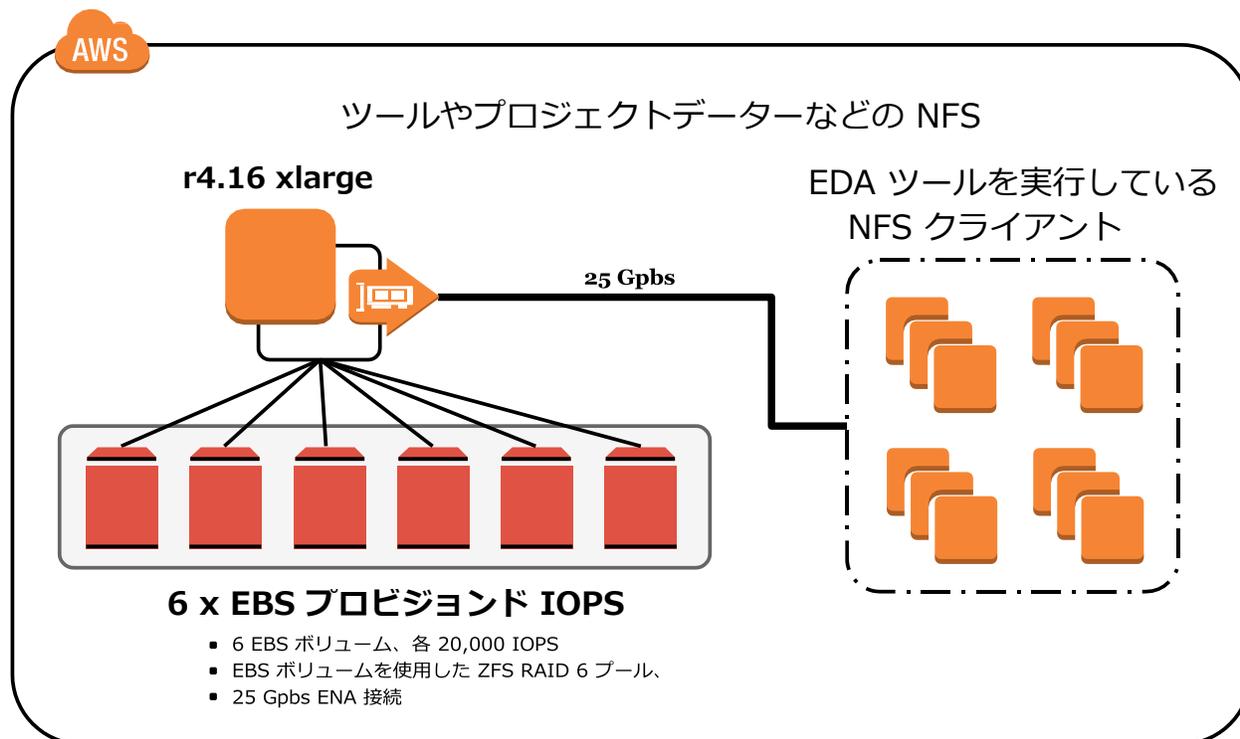
日付	説明
2018 年 9 月	2018 年の更新
2017 年 10 月	初版発行

## 付録 A - ストレージの最適化

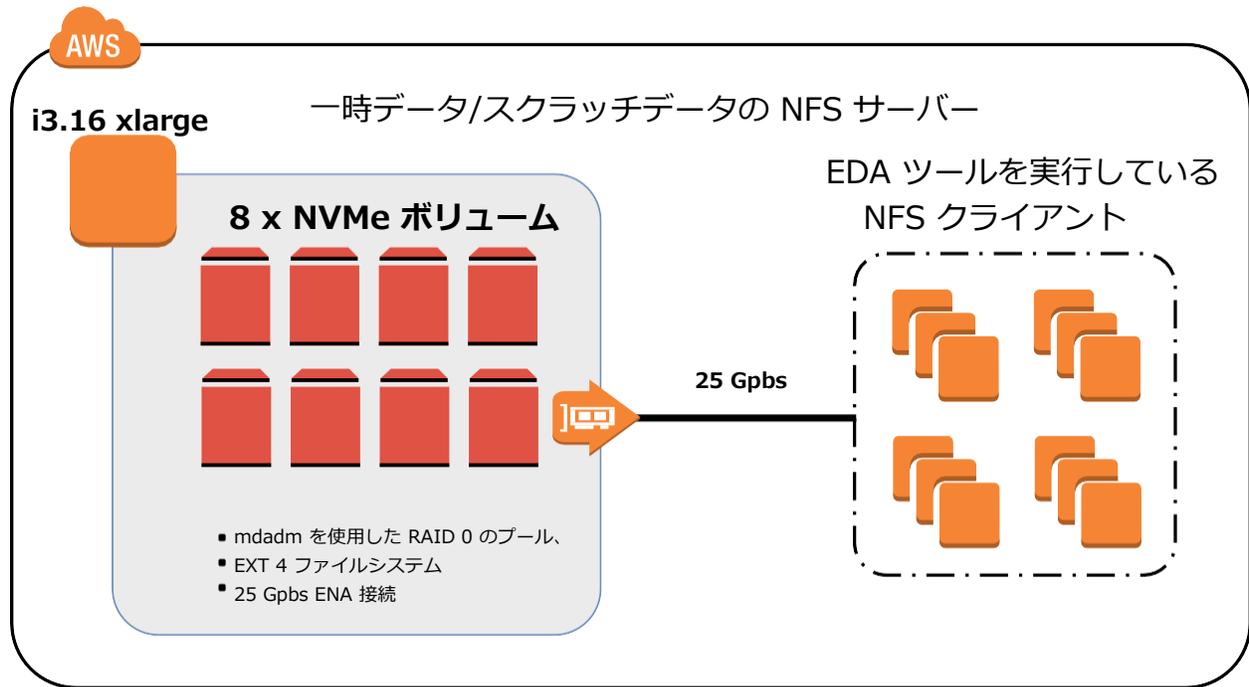
AWS にはさまざまなストレージオプションがあり、高レベルのものも含まれています。半導体のワークロードは共有ストレージに依存していることから、NFS サーバーの構築が EDA ツールを実行する最初のステップとも言えます。このセクションでは、多くのワークロードに適切なパフォーマンスを実現できる可能性のある 2 つの NFS アーキテクチャについて説明します。

### NFS ストレージ

#### 1.75 GB/秒の 75,000 IOPS 対応の NFS サーバー



## 2.5 GB/ 秒の 100,000 IOPS 対応の NFS サーバー



## 付録 B - リファレンスアーキテクチャ

下記の図は、AWS で 拡張が可能な EDA のコンピューティング環境向けの一般的なアーキテクチャです。この設計では、以下の主要なインフラストラクチャコンポーネントを提供します。

- 拡張用の Amazon EC2 Auto Scaling グループ
- AWS の専用回線接続であるAWS Direct Connect
- リモートデスクトップ用の Amazon Linux WorkSpaces
- Amazon EC2 ベースのコンピューティング、ライセンス、スケジューラインスタンス
- コンピュートインスタンス間でファイルシステムを共有するための Amazon EC2 ベースの NFS サーバーと Amazon EFS

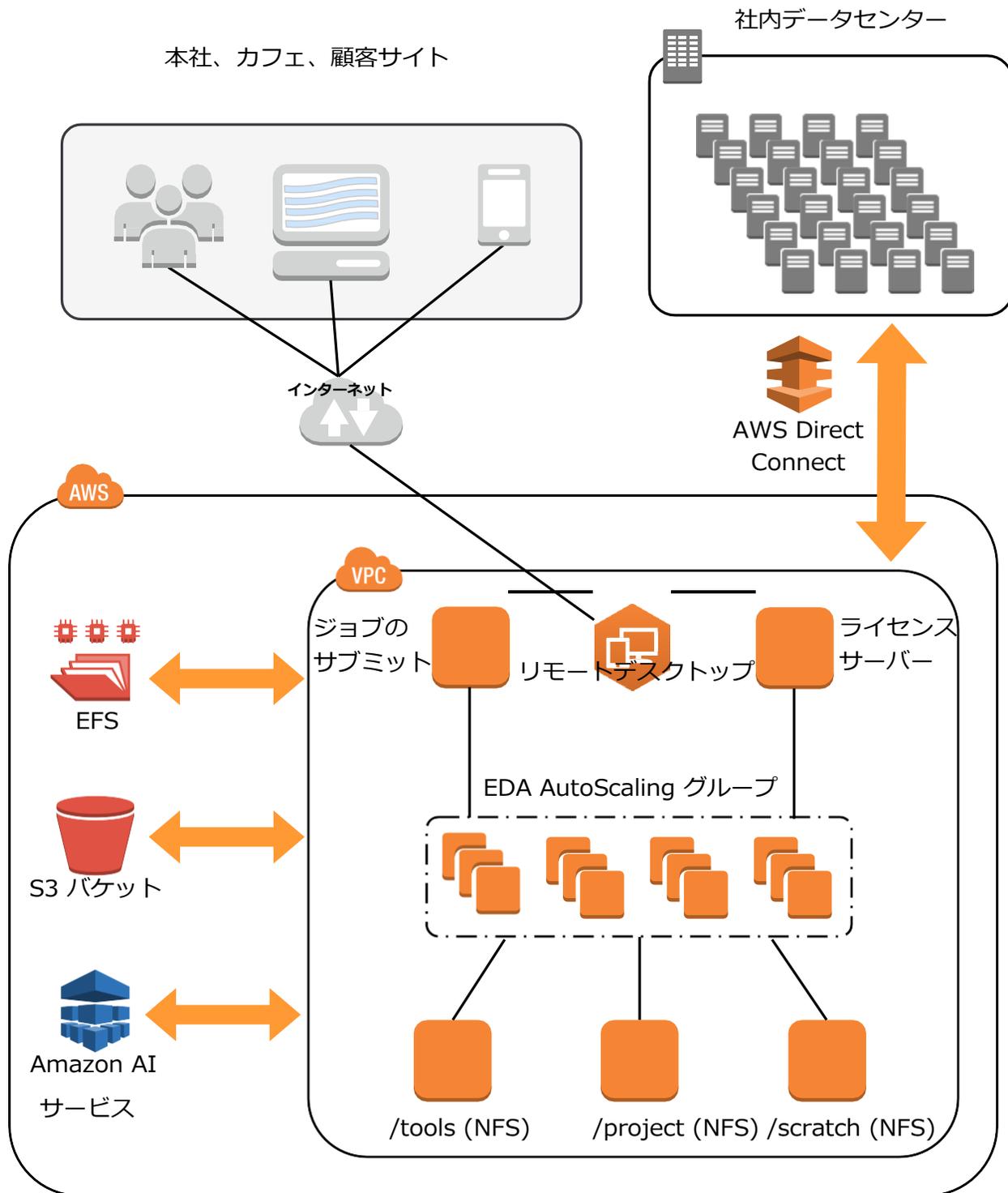


図 5 : AWSでのアーキテクチャ EDA

## 付録 C - Linux カーネルコマンドラインの更新

### /etc/default/grub ファイルを使ったシステムの更新

1. 適切なエディタで /etc/default/grub ファイルを開きます。

```
$ sudo vim/etc/default/grub
```

2. GRUB\_CMDLINE\_LINUX\_DEFAULT の行を編集し、必要な変更を加えます。例

```
GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8  
net.ifnames=0 biosdevname=0 nvme_core.io_timeout=4294967295  
intel_idle.max_cstate=1"
```

3. ファイルを保存し、エディタを終了します。
4. 次のコマンドを実行し、ブート構成を再構築します。

```
$ grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. インスタンスを再起動し、新しいカーネルオプションを有効にします。

```
$ sudo reboot
```

## /boot/grub/grub.conf ファイルを使ったシステムの更新

6. 任意のテキストエディタで /boot/grub/grub.conf ファイルを開きます。

```
$ sudo vim /boot/grub/grub.conf
```

7. 次のようにカーネル行を編集します (分かりやすいように一部の情報は除外しています)。

```
# created by imagebuilder
default=0
timeout=1
hiddenmenu

title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
root (hd 0,0)
kernel /boot/vmlinuz-ver.amzn1.x86_64 <other_info> intel_idle.max_cstate=1
initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
```

8. ファイルを保存し、エディタを終了します。
9. インスタンスを再起動し、新しいカーネルオプションを有効にします。

```
$ sudo reboot
```

## カーネル行の検証

dmesg または /proc/cmdline のカーネルのコマンド行を実行して、設定を検証します。

```
$ dmesg | grep 「Kernel Command Line」
[    0.000000] Kernel command line: root=LABEL=/ console=tty1
console=ttyS0 maxcpus=18 xen_nopvspin=1

$ cat/proc/cmdline
root=LABEL=/ console=tty1 console=ttyS0 maxcpus=18 xen_nopvspin=1
```

## 注釈

- <sup>1</sup> <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ssd-instance-store.html>
- <sup>2</sup> [http://docs.aws.amazon.com/directoryservice/latest/admin-guide/directory\\_simple\\_ad.html](http://docs.aws.amazon.com/directoryservice/latest/admin-guide/directory_simple_ad.html)