



ゲノム解析のための **FlexPod** FlexPod

NetApp
October 30, 2025

目次

ゲノム解析のための FlexPod	1
TR-4911 : 『FlexPod Genomics』	1
適用範囲	2
対象者	2
FlexPod に導入された病院機能	2
ゲノムワークロードを FlexPod に導入するメリット	3
解決策インフラのハードウェアコンポーネントとソフトウェアコンポーネント	8
ゲノム - GATK のセットアップと実行	12
サンプルからバリエーションの識別、注釈、および予測まで	12
ゲノム解析ツールキット (GATK)	14
FlexPod セットアップ	14
GATK のセットアップと実行	18
jar ファイルを使用して GATK を実行するための出力	22
./GATK スクリプトを使用して GATK を実行するための出力	25
クロムウェルエンジンを使用した GATK 実行用出力	27
GPU セットアップ	31
まとめ	40
追加情報の参照先	41
バージョン履歴	42

ゲノム解析のための FlexPod

TR-4911 : 『 FlexPod Genomics 』

ネットアップ、JayaKishore Esankula

医療や生命科学のゲノムよりも重要な医療分野がいくつかあり、ゲノム研究は医師や看護師にとって重要な臨床ツールとなりつつあります。ゲノムと医療画像およびデジタル病理学を組み合わせることで、患者の遺伝子が治療プロトコルによってどのように影響を受けるかを理解できます。医療におけるゲノム研究の成功は、データの大規模な相互運用性にますます左右されています。最終的な目標は、膨大な量の遺伝子データを理解し、臨床的に関連性のある相関と変異を特定して診断を改善し、精密医療を現実にすることです。ゲノム研究では、病気の発生源、疾患の進化、どの治療法や戦略が効果的かを理解することができます。明らかに、ゲノムには、予防、診断、治療にまたがる多くのメリットがあります。医療機関は、次のようないくつかの課題に取り組んでいます。

- ケア品質の向上
- 価値に基づく治療
- データの急増
- 精密医学
- パンダ
- ウェアラブル機器、リモートモニタリング、ケア
- サイバーセキュリティ

標準化された臨床経路と臨床プロトコルは、現代医学の重要な要素の 1 つです。標準化の重要な側面の 1 つは、医療記録だけでなくゲノムデータに対しても、医療提供者間の相互運用性です。医療機関は、個人のゲノムデータや関連する医療記録を患者が所有するのではなく、ゲノムデータの所有権を放棄するのでしょうか？

相互運用可能な患者データは、データの急増に対応する原動力の 1 つである精密医療を実現するための鍵となります。精密医療の目的は、健康維持、疾病予防、診断、治療ソリューションをより効果的かつ正確に行うことです。

データの増加率は急激に上昇しています。2021 年 2 月上旬に、米国の研究所では、1 週間に約 8、000 系統の新型コロナウイルス感染症のシーケンスが確認されました。ゲノム配列の数は 2021 年 4 月までに週 29,000 に増加しました。完全に配列されたヒトゲノムのサイズは約 125GB です。したがって、1 週間に 29,000 個のゲノム配列を持つゲノムの保管データは、1 年間で 180 ペタバイトを超えることになります。さまざまな国がゲノム疫学にリソースを投入し、ゲノム監視を改善し、世界的な健康問題の次の波に備えるよう取り組んでいます。

ゲノム研究のコスト削減により、遺伝子検査や研究はかつてないほどのスピードで進められています。3 つの PS は、コンピュータのパワー、データのプライバシー、医療のパーソナライズというターニングポイントにあります。2025 年には、研究者らは、ヒトゲノムの配列が 1 億～200 億個になると予測している。ゲノム研究を効果的かつ価値ある提案にするには、ゲノム機能が医療ワークフローのシームレスな一部である必要があります。患者の訪問時に、簡単にアクセスして実行可能である必要があります。患者の電子カルテデータを患者のゲノムデータと統合することも同様に重要です。FlexPod のような最先端の統合インフラの出現により、組織はゲノム機能を医師、看護師、診療所のマネージャーの日常的なワークフローに導入できるようになります。

した。FlexPod プラットフォームの最新情報については、こちらを参照してください "[FlexPod Datacenter with Cisco UCS X Series White Paper](#)』を参照してください"。

ゲノム研究の真の価値は、患者のゲノムデータに基づく精密な医療と個別化された治療計画です。過去に臨床医とデータサイエンティストの相乗効果が生まれたことはありません。ゲノム研究は、これまでの最新技術革新の恩恵を受けています。また、医療機関と業界の技術リーダーとの真のパートナーシップも享受しています。

学術医療センターやその他の医療および生命科学機関は、ゲノム科学の中心的研究拠点（COE）を確立するために十分に活用されています。博士によるCharlie Gersbach、Dr.グレッグ・クロフォード博士Duke UniversityのTim E Reddy氏は、「単純なバイナリスイッチでは遺伝子のオン/オフが行われていないことはわかっていますが、複数の遺伝子規制切り替えが連携して機能する結果です。また、「これらのゲノムの部分は、いずれも独立して機能しないと判断しました。ゲノムは非常に複雑なウェブで、進化してきました」（"[参照（Ref）](#)）。

ネットアップとCiscoは、10年以上にわたってFlexPodプラットフォームをさらに強化してきました。すべてのお客様からのフィードバックは、FlexPodのバリューストリームと機能セットに耳を傾け、評価し、結び付けられます。この継続的なフィードバック、コラボレーション、改善、お祝いのループは、FlexPodを信頼できる統合インフラストラクチャプラットフォームとして世界中で差別化します。シンプル化され、一から設計されたこのプラットフォームは、医療機関にとって最も信頼性が高く、堅牢で汎用性が高く、即応性に優れたプラットフォームです。

適用範囲

FlexPod コンバージドインフラプラットフォームを使用すると、医療機関は1つ以上のゲノム関連ワークロードと、他の臨床/非臨床的な医療アプリケーションをホストできます。このテクニカルレポートでは、FlexPodプラットフォームの検証時にGATKと呼ばれる、オープンソースの業界標準ゲノムツールを使用しています。ただし、ゲノム解析やGATKについて詳しくは、このドキュメントでは扱いません。

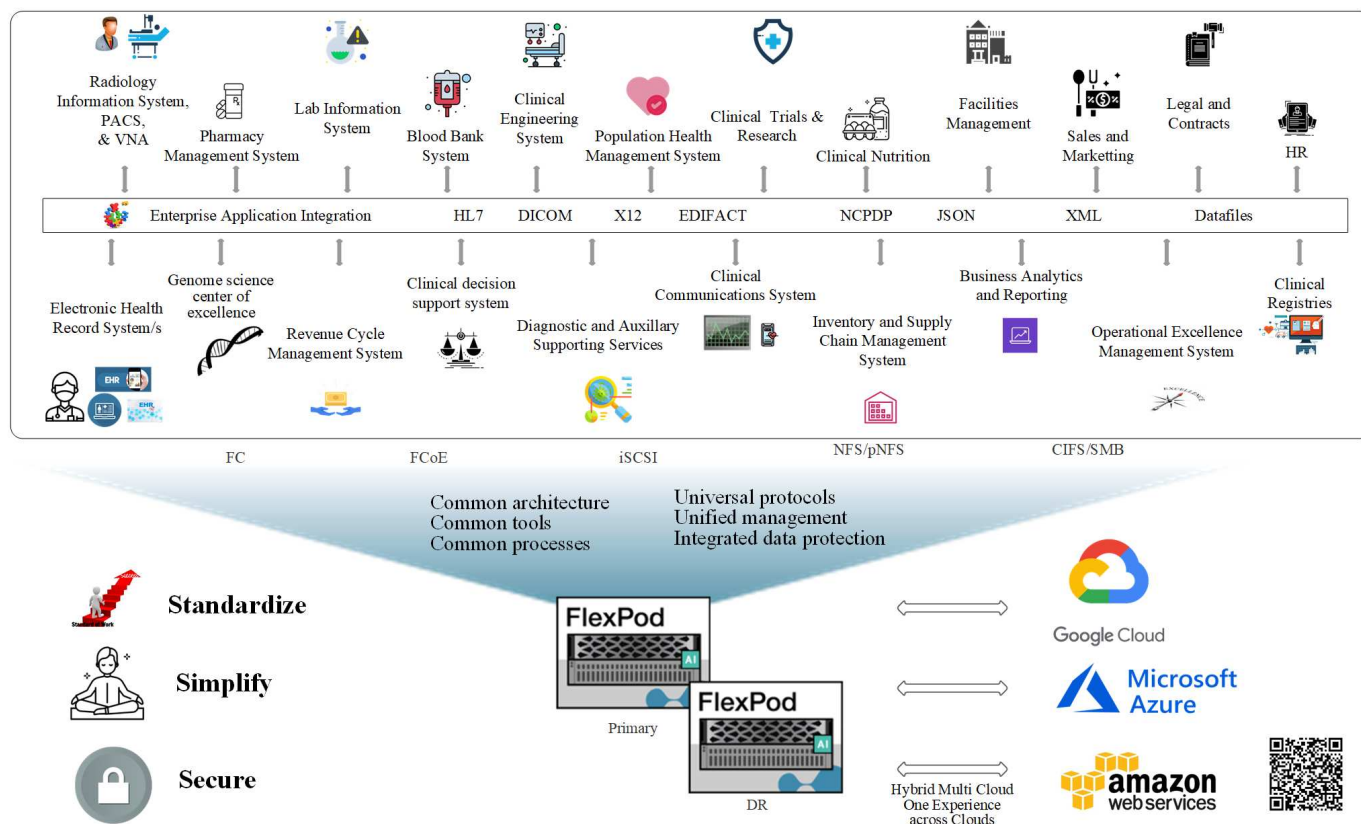
対象者

本ドキュメントは、医療業界の技術リーダー、Ciscoとネットアップのパートナーソリューションエンジニア、およびプロフェッショナルサービス担当者を対象としています。本ドキュメントは、コンピューティングとストレージのサイジングの概念に加え、医療の脅威、医療セキュリティ、医療ITシステム、Cisco UCS、ネットアップストレージシステムに関する技術的な知識があることを前提としています。

FlexPod に導入された病院機能

一般的な病院にはITシステムが多様化しています。このようなシステムの大半はベンダーから購入されますが、社内の病院システムによって構築されるものはほとんどありません。そのため、病院システムはデータセンターの多様なインフラ環境を管理する必要があります。病院がシステムをFlexPodなどの統合インフラプラットフォームに統合すれば、データセンターの運用を標準化できます。FlexPodを使用すると、医療機関は臨床システムと非臨床システムを同じプラットフォームに実装できるため、データセンターの運用を統合できます。

Hospital capabilities deployed on a FlexPod



"次は、ゲノムワークロードを FlexPod に導入するメリットです。"

ゲノムワークロードを FlexPod に導入するメリット

"前へ：はじめに。"

このセクションでは、FlexPod コンバインドインフラプラットフォームでゲノミクスワークロードを実行する利点の概要を説明します。では、病院の機能について簡単に説明しましょう。次のビジネスアーキテクチャビューは、ハイブリッドクラウド対応の FlexPod コンバインドインフラストラクチャプラットフォームに導入された病院の機能を示しています。

- * 医療でサイロ化を避ける。* 医療のサイロ化は非常に大きな懸念事項です。多くの場合、部門は選択したものではなく、進化によって組織的に孤立した独自のハードウェアとソフトウェアのセットにサイロ化しています。放射線科、心臓病、EHR、ゲノム解析など分析、収益サイクル、その他の部門は、個々の専用ソフトウェアとハードウェアで構成されます。医療機関では、ハードウェアとソフトウェアの資産を管理するための IT プロフェッショナルが限られています。このような転換点は、非常に多様なハードウェアとソフトウェアを管理することが求められている場合に生じます。ベンダーによって医療機関にもたらされた矛盾する一連のプロセスによって、不均質性が悪化します。
- * 小さい始め、育つ。* GATK の用具キットは CPU の実行のために調整される FlexPod のような最もよいスイートルームのプラットフォーム。FlexPod を使用すると、ネットワーク、コンピューティング、ストレージの拡張性を個別に拡張できます。ゲノム機能と環境の拡大に合わせて小規模な構成から始め、拡張できます。医療機関は、ゲノムワークロードを実行するために特化したプラットフォームに投資する必要がありません。代わりに、FlexPod のような汎用性の高いプラットフォームを活用して、同じプラットフ

フォーム上でゲノミクスとゲノム以外のワークロードを実行できます。たとえば、小児部門がゲノム機能の実装を検討している場合、IT リーダーは既存の FlexPod インスタンスでコンピューティング、ストレージ、ネットワークをプロビジョニングできます。ゲノムビジネスユニットの成長に伴い、医療機関は必要に応じて FlexPod プラットフォームを拡張できます。

- * 単一のコントロールパネルと比類のない柔軟性。* Cisco Intersight は、アプリケーションとインフラストラクチャをブリッジすることで、IT 運用を大幅に簡易化し、ベアメタルサーバやハイパーバイザからサーバレスアプリケーションまでの可視化と管理を実現し、コストを削減し、リスクを軽減します。このユニファイド SaaS プラットフォームは、オープン API 設計を採用しており、サードパーティのプラットフォームやツールとネイティブに統合されています。さらに、モバイルアプリを使用して、データセンター運用チームからオンサイトまたは場所を問わず管理を行うことができます。

ユーザは、Intersight を管理プラットフォームとして活用することで、目に見える形での価値をすばやく引き出すことができます。多くの日常的な手作業の自動化を可能にする Intersight は、エラーを解消し、日常業務を簡易化します。さらに、Intersight の高度なサポート機能により、導入者は問題に先手を打つことができ、問題の解決を加速できます。企業がアプリケーションインフラに費やす時間とコストを大幅に削減し、コアビジネスの開発にかける時間を増やしています。

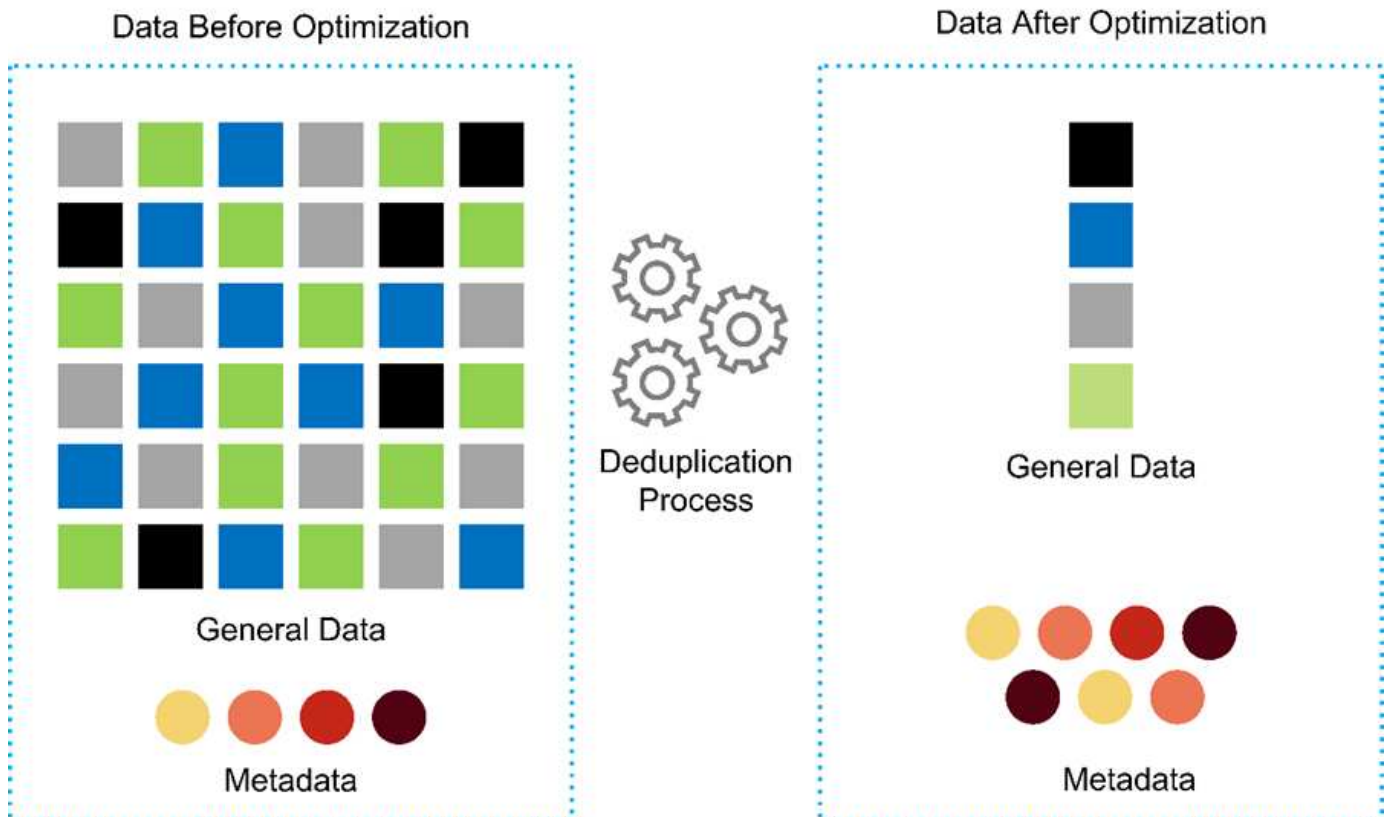
Intersight の管理と FlexPod の拡張性に優れたアーキテクチャを活用することで、複数のゲノムワークロードを単一の FlexPod プラットフォームで実行できるようになり、利用率が向上し、総所有コスト（TCO）が削減されます。FlexPod では、ネットアップの小規模な FlexPod Express から始めて、大規模な FlexPod データセンター実装まで拡張できるため、柔軟なサイジングが可能です。Cisco Intersight に組み込まれているロールベースのアクセス制御機能により、医療機関は堅牢なアクセス制御メカニズムを実装して、個別のインフラスタックを必要としないようにすることができます。医療機関内の複数のビジネスユニットが、ゲノム研究を主要な中核コンピテンシーとして活用できます。

最終的に FlexPod は、IT 運用を簡易化し、運用コストを削減します。IT インフラストラクチャ管理者は、臨床医の革新を支援するタスクに集中できるため、常に最新の状態に維持することはできません。

- * 検証済みの設計と保証された成果。* FlexPod の設計および導入ガイドは、再現可能であることが検証されており、FlexPod を確実に導入するために必要な包括的な構成の詳細と業界のベストプラクティスが記載されています。Cisco とネットアップの検証済み設計ガイド、導入ガイド、アーキテクチャを活用すれば、医療機関やライフサイエンス部門が、検証済みで信頼性の高いプラットフォームを最初から導入する際に推測に頼ることがなくなります。FlexPod を使用すると、導入時間を短縮し、コスト、複雑さ、リスクを軽減できます。FlexPod 検証済みの設計と導入ガイドでは、さまざまなゲノムワークロードに最適なプラットフォームとして FlexPod を確立しています。
- * 革新性と俊敏性。* FlexPod は Epic、Cerner、Meditech、Agfa、GE、Philips などの画像処理システムなどの EHR によって理想的なプラットフォームとして推奨されています。詳細については、を参照してください ["EPIC 名誉の転がる"](#) ターゲットとなるプラットフォームアーキテクチャについては、Epic userweb を参照してください。ゲノム解析の実行 ["FlexPod"](#) 医療機関は、即応性を備えた革新的なビジネスを継続できます。FlexPod を導入することで、組織の変化を自然に実現できます。医療機関が FlexPod プラットフォームを標準化すると、IT エキスパートは時間、労力、リソースをプロビジョニングしてイノベーションを推進できるようになり、エコシステムのニーズに合わせた即応性が実現します。
- * データの制約を解放。* ONTAP コンバージドインフラプラットフォームと NetApp FlexPod ストレージシステムを使用すると、ゲノムデータを 1 つのプラットフォームから幅広いプロトコルで大規模に利用およびアクセスできます。FlexPod と NetApp ONTAP は、シンプルでわかりやすく、強力なハイブリッドクラウドプラットフォームです。NetApp ONTAP を基盤とするデータファブリックは、サイト間、物理的な境界を超え、アプリケーション間でデータを結び付けます。データファブリックは、Data-Centric の世界におけるデータ主体の企業向けに構築されています。データは複数の場所に作成されて使用されるため、多くの場合、他の場所、アプリケーション、インフラとの利用や共有が必要になります。そのため、一貫性のある統合された管理方法が必要です。FlexPod を導入することで、IT チームの管理が容易になり、増え続ける IT の複雑さが軽減されます。
- * セキュアマルチテナンシー。* FlexPod は FIPS 140-2 準拠の暗号モジュールを使用しているため、セキ

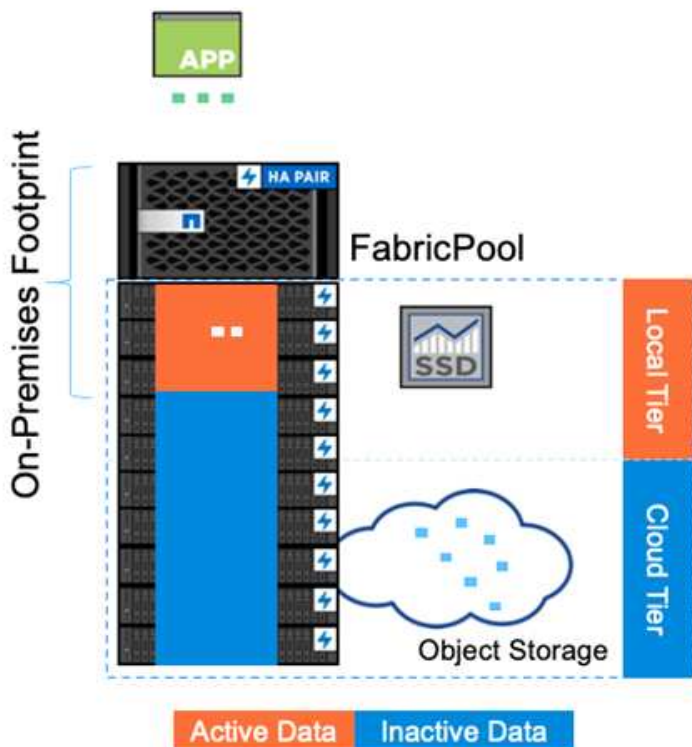
セキュリティを後からではなく基本要素として実装できます。FlexPod を使用すると、プラットフォームの規模に関係なく、単一のコンバインドインフラプラットフォームからセキュアマルチテナンシーを実装できます。マルチテナンシーと QoS で FlexPod を保護することで、ワークロードの分離と利用率の最大化を実現します。これにより、使用率の低い可能性のある特殊なプラットフォームに設備が固定され、管理に特殊なスキルセットが必要になるのを回避できます。

- * ストレージ効率化 * ゲノミクスには、基盤となるストレージに業界をリードするストレージ効率化機能が必要です。重複排除（インラインおよびオンデマンド）、データ圧縮、データコンパクション（など）などのネットアップの Storage Efficiency 機能を使用すると、ストレージコストを削減できます ["参照（Ref）"](#)）。ネットアップの重複排除機能は、FlexVol ボリューム内でブロックレベルの重複排除を実行します。重複排除機能は、基本的に、重複ブロックを削除して、FlexVol ボリューム内で一意のブロックのみを保存します。重複排除は非常にきめ細かな方法で機能し、FlexVol ボリュームのアクティブファイルシステムで機能します。次の図に、ネットアップの重複排除機能の概要を示します。重複排除機能はアプリケーションに対して透過的です。したがって、ネットアップシステムを使用するすべてのアプリケーションのデータに対して重複排除を実行できます。ボリューム重複排除はインラインプロセスおよびバックグラウンドプロセスとして実行できます。CLI、NetApp ONTAP System Manager、または NetApp Active IQ Unified Manager から自動で実行、スケジュール設定、または手動で実行するように設定することができます。



- * ゲノムの相互運用性を実現。* ONTAP FlexCache は、ファイル配信を簡素化し、WAN のレイテンシを低減し、WAN 帯域幅コスト（["参照（Ref）"](#)）。ゲノム変異の同定およびアノテーションにおける重要な活動の 1 つに、臨床医間のコラボレーションがあります。ONTAP FlexCache テクノロジーは、コラボレーションする臨床医が異なる地域にいる場合でも、データのスループットを向上させます。一般的な *。BAM ファイルのサイズ（1 GB ～ 100 GB）を考えると、基盤となるプラットフォームが異なる地域の臨床医がファイルを使用できるようにすることが重要です。FlexPod と ONTAP FlexCache を併用することで、ゲノムデータとアプリケーションをマルチサイトに対応できます。その結果、世界中に分散している研究者が、低レイテンシと高スループットを実現しながらシームレスに連携できるようになります。ゲノム研究アプリケーションをマルチサイト環境で実行している医療機関は、データファブリックを使用してスケールアウトを実施し、管理性とコスト、スピードのバランスを取ることができます。

- *ストレージ・プラットフォームのインテリジェントな使用*FlexPodとONTAP自動階層化およびNetApp FabricPoolテクノロジーを組み合わせることで、データ管理が簡易化されます。FabricPool は、パフォーマンス、効率、セキュリティ、保護を犠牲にすることなくストレージコストを削減します。FabricPool は、エンタープライズアプリケーションに対して透過的であり、アプリケーションインフラを再構築することなくストレージの TCO を削減することで、クラウドの効率性を活用します。FlexPod は、FabricPool のストレージ階層化機能を活用して、ONTAP フラッシュストレージをより効率的に使用できます。詳細については、を参照してください "[FlexPod with FabricPool の略](#)"。次の図は、FabricPool とその利点の概要を示しています。



Automatic tiering
Zero-touch management
Preserves file system
Lower cost of ownership
Choice of object tier locations



- *バリエーション分析とアノテーションの高速化。* FlexPod プラットフォームの導入と運用開始にかかる時間が短縮されます。FlexPod プラットフォームでは、データを低レイテンシとスループットで大規模に利用できるようにすることで、医療従事者によるコラボレーションが可能になります。相互運用性が向上することで、イノベーションが医療機関は、ゲノムワークロードと非ゲノムワークロードを並行して実行できるため、ゲノム解析の開始に特化したプラットフォームを必要としません。

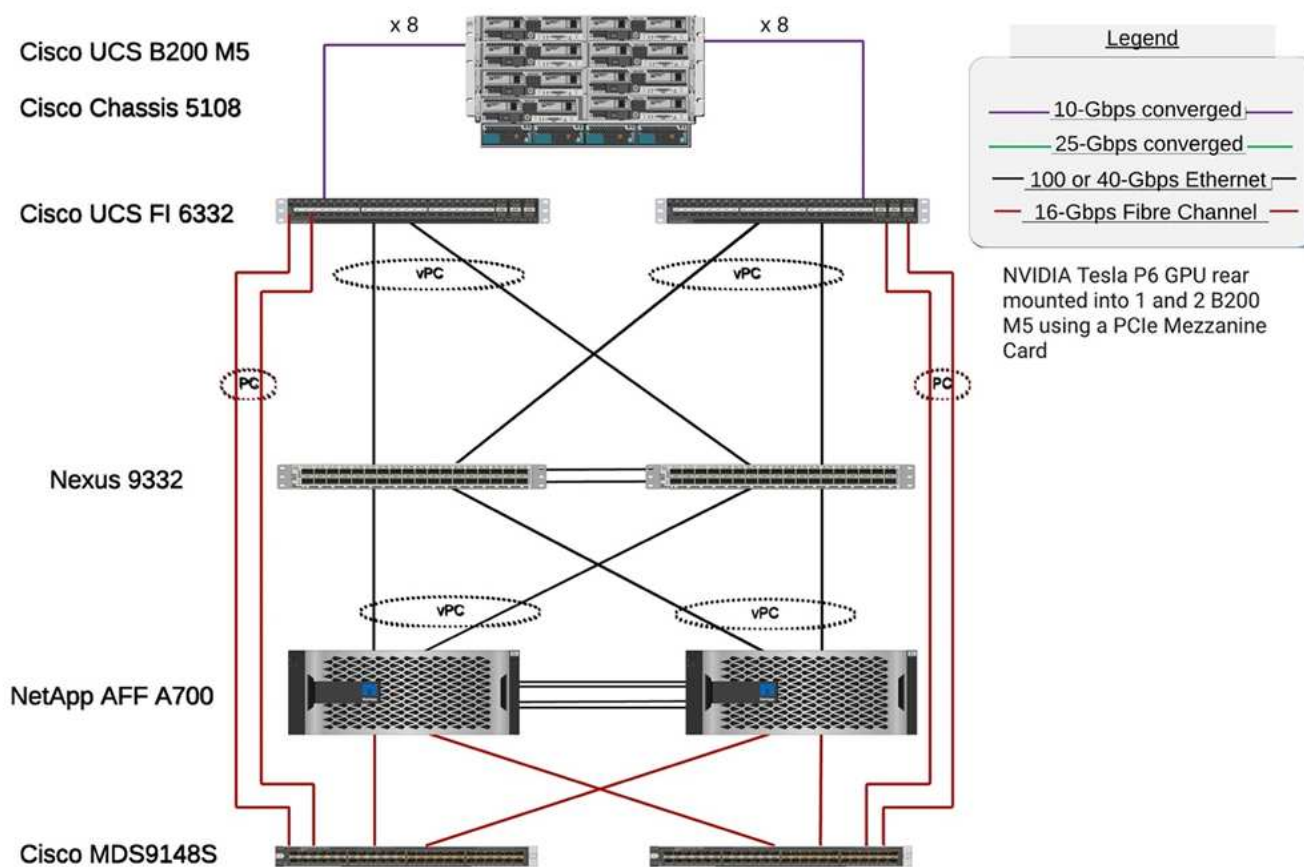
FlexPod ONTAP では、ストレージプラットフォームに最新の機能を定期的に追加しています。FlexPod データセンターは、FC-NVMe を導入するための最適な共有インフラ基盤であり、必要なアプリケーションにハイパフォーマンスなストレージアクセスを提供します。FC-NVMe は進化し、高可用性、マルチパス、およびオペレーティングシステムの追加サポートが組み込まれています。FlexPod は、このような機能をサポートするために必要な拡張性と信頼性を備えたプラットフォームに最適です。エンドツーエンド NVMe で I/O を高速化した ONTAP により、ゲノム解析を高速化 ("[参照 \(Ref\)](#) ") 。

ゲノム配列データは大きなファイルサイズを生成します。これらのファイルをバリエーション分析装置で利用できるようにすることで、サンプルの収集からバリエーションの注釈までにかかる総時間を短縮することが重要です。ストレージアクセスおよびデータ転送プロトコルとして使用される NVMe (Non-Volatile Memory Express) は、かつてないレベルのスループットと最速の応答時間を実現します。FlexPod は、PCI Express Bus (PCIe ; PCI Express バス) を介してフラッシュストレージにアクセスしながら、NVMe プロトコルを導入します。PCIe により、数万のコマンドキューの実装が可能になり、並列化とス

ループットが向上します。ストレージからメモリまで 1 つのプロトコルでデータアクセスが高速化されます。

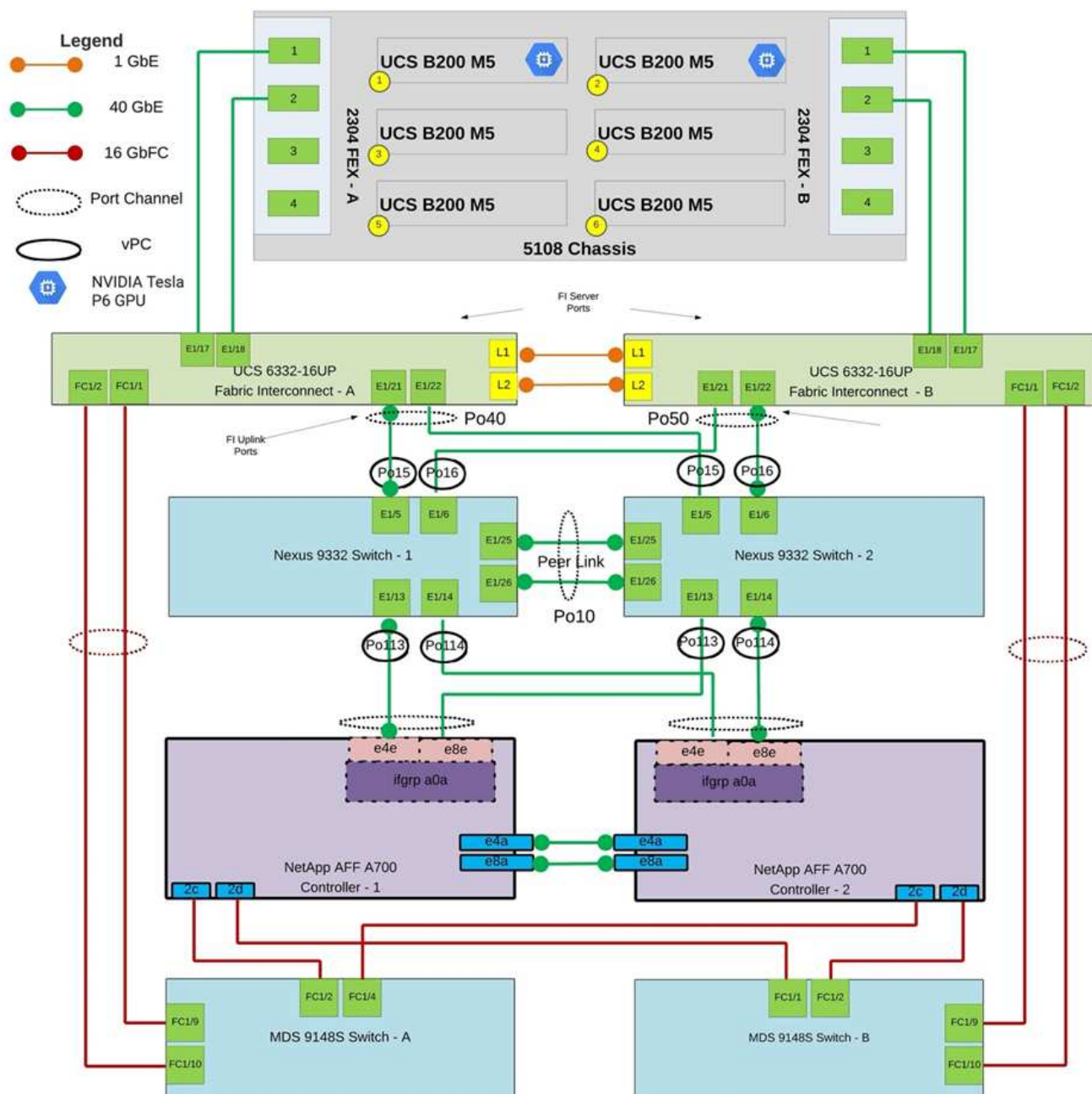
- * 臨床研究の俊敏性を徹底的に高めています。 * 柔軟で拡張可能なストレージ容量とパフォーマンスにより、医療研究機関は柔軟でジャストインタイム（JIT）方式で環境を最適化できます。コンピューティングインフラとネットワークインフラのストレージを分離 FlexPod することで、システムを停止することなくスケールアップとスケールアウトが可能です。Cisco Intersight を使用すると、FlexPod プラットフォームの管理に組み込みの自動ワークフローとカスタムの自動ワークフローの両方を利用できます。Cisco Intersight のワークフローにより、医療機関はアプリケーションのライフサイクル管理時間を短縮できます。学術医療センターでは、患者データを匿名化して研究インフォマティクスやセンターで高品質な情報を提供する場合、IT 部門は Cisco Intersight FlexPod のワークフローを活用して、セキュアなデータバックアップ、クローニング、リストアを数時間ではなく数秒で実行できます。NetApp Trident と Kubernetes を使用すると、IT 部門は新しいデータサイエンティストをプロビジョニングし、臨床データをわずか数分でモデル開発に利用できます。しかも数秒で完了することもあります。
- * ゲノムデータを保護。 * NetApp SnapLock は、消去や書き換えが不可能な状態でファイルを保存し、コミットできる特殊な用途に対応しています。FlexVol ボリュームに保存されているユーザーの本番データは、NetApp SnapMirror または SnapVault テクノロジーを使用して、SnapLock ボリュームにミラーリングまたは保存できます。SnapLock ボリューム内のファイル、ボリューム自体、およびホストアグリゲートは、保持期間が終了するまで削除できません。ONTAP FPolicy ソフトウェアを使用している組織では、特定の拡張子のファイルに対する処理を禁止することで、ランサムウェア攻撃を防止できます。FPolicy イベントは、特定のファイル操作に対してトリガーできます。イベントはポリシーに関連付けられており、ポリシーは使用する必要があるエンジン呼び出します。ポリシーにはランサムウェアを含む可能性のある一連のファイル拡張子を設定できます。拡張子が許可されていないファイルで許可されていない操作を実行しようとする、FPolicy によりその操作が実行されなくなります。([参照 \(Ref\)](#))。
- * FlexPod 共同サポート * ネットアップと Cisco は、FlexPod コンバインドインフラに固有のサポート要件を満たす、拡張性と柔軟性に優れた強力なサポートモデルである FlexPod 共同サポートを確立しました。このモデルでは、ネットアップと Cisco が提供する経験、リソース、およびテクニカルサポートの専門知識を組み合わせ、問題の発生場所に関係なく、FlexPod のサポート問題を特定して解決するための合理的なプロセスを提供します。次の図に、FlexPod 共同サポートモデルの概要を示します。お客様は、問題を所有する可能性のあるベンダーに連絡し、Cisco とネットアップは協力して解決するように依頼します。Cisco とネットアップには、複数の企業にわたるエンジニアリングチームと開発チームがあり、これらのチームが協力して問題を解決します。このサポートモデルにより、翻訳中の情報の損失を削減し、信頼性を高め、ダウンタイムを削減できます。

FlexPod for Genomics



次の図は、FlexPod のケーブル配線の詳細を示しています。

FlexPod for Genomics



次の表に、FlexPod で有効にする GATK テスト中に使用されるハードウェアコンポーネントを示します。はこちらです "[NetApp Interoperability Matrix Tool で確認できます](#)" (IMT) および "[シスコハードウェア互換性リスト \(HCL\)](#)"。

レイヤー (Layer)	製品ファミリー	数量とモデル	詳細
コンピューティング	Cisco UCS 5108 シャーシ	1 または 2	
	Cisco UCS ブレードサーバ	B200 M5 × 6	それぞれに、20 コア以上、2.7GHz、および 128-384GB RAM を 2 個搭載しています

レイヤー（Layer）	製品ファミリー	数量とモデル	詳細
	Cisco UCS 仮想インターフェイスカード（VIC）	Cisco UCS 1440	を参照してください
	Cisco UCS ファブリックインターコネクト × 2	6332	-
ネットワーク	Cisco Nexus スイッチ	Cisco Nexus 9332 × 2	-
ストレージネットワーク	SMB / CIFS、NFS、または iSCSI プロトコル経由のストレージアクセス用の IP ネットワーク	上記と同じネットワークスイッチ	-
	FC 経由のストレージアクセス	Cisco MDS 9148S × 2	-
ストレージ	NetApp AFF A700 オールフラッシュストレージシステム	1 クラスタ	2 ノードクラスタ
	ディスクシェルフ	DS224C または NS224 ディスクシェルフ × 1	24 本のドライブをフル装備
	SSD の場合	容量が 24、2TB 以上	-

この表は、インフラストラクチャソフトウェアを示しています。

ソフトウェア	製品ファミリー	バージョンまたはリリース	詳細
様々	Linux の場合	RHEL 8.3	-
	Windows の場合	Windows Server 2012 R2（64 ビット）	-
	NetApp ONTAP	ONTAP 9.8 以降	-
	Cisco UCS ファブリックインターコネクト	Cisco UCS Manager 4.1 以降	-
	Cisco Ethernet 3000 または 9000 シリーズスイッチ	9000 シリーズの場合、7.0(3) i7(7) 以降（3000 シリーズ用）、9.2(4) 以降	-
	Cisco FC : Cisco MDS 9132T	8.4(1a) 以降	-
	ハイパーバイザー	VMware vSphere ESXi 7.0	-
ストレージ	ハイパーバイザー管理システム	VMware vCenter Server 7.0（vCSA）以降	-
ネットワーク	NetApp Virtual Storage Console（VSC）	VSC 9.7 以降	-
	NetApp SnapCenter	SnapCenter 4.3 以降	-

ソフトウェア	製品ファミリー	バージョンまたはリリース	詳細
	Cisco UCS Manager の略	4.1 (3c) 以降	
ハイパーバイザー	ESXi		
管理	ハイパーバイザー管理システム VMware vCenter Server 7.0 (vCSA) 以降		
	NetApp Virtual Storage Console (VSC)	VSC 9.7 以降	
	NetApp SnapCenter	SnapCenter 4.3 以降	
	Cisco UCS Manager の略	4.1 (3c) 以降	

"次は、Genomics - GATK のセットアップと実行です。"

ゲノム - GATK のセットアップと実行

"以前：解決策インフラのハードウェアコンポーネントとソフトウェアコンポーネント。"

国立ヒトゲノム研究所によると **"NHGRI"**、「ゲノムとは、人の遺伝子 (ゲノム) のすべての研究であり、これらの遺伝子相互作用や人の環境との相互作用を含みます。」

に従って **"NHGRI"**「デオキシリボヌクレオ酸 (DNA) は、ほぼすべての生物の活動を開発し、誘導するために必要な指示を含む化学化合物です。DNA 分子は、二重らせんと呼ばれる 2 つのツイスト、ペアストランドで構成されています。」「生物の DNA の完全なセットは、ゲノムと呼ばれています。」

配列決定は DNA の鎖の塩基の正確な順序を決定するプロセスである。現在使用されている最も一般的なシーケンスタイプの 1 つは、合成による順序付けと呼ばれます。この技術では、蛍光信号の放射を使用して塩基を並べます。研究者は DNA シーケンシングを使用して、遺伝子変異や、人がまだ初期段階にある間に疾患の発症または進行に関与する可能性のある突然変異を検索することができる。

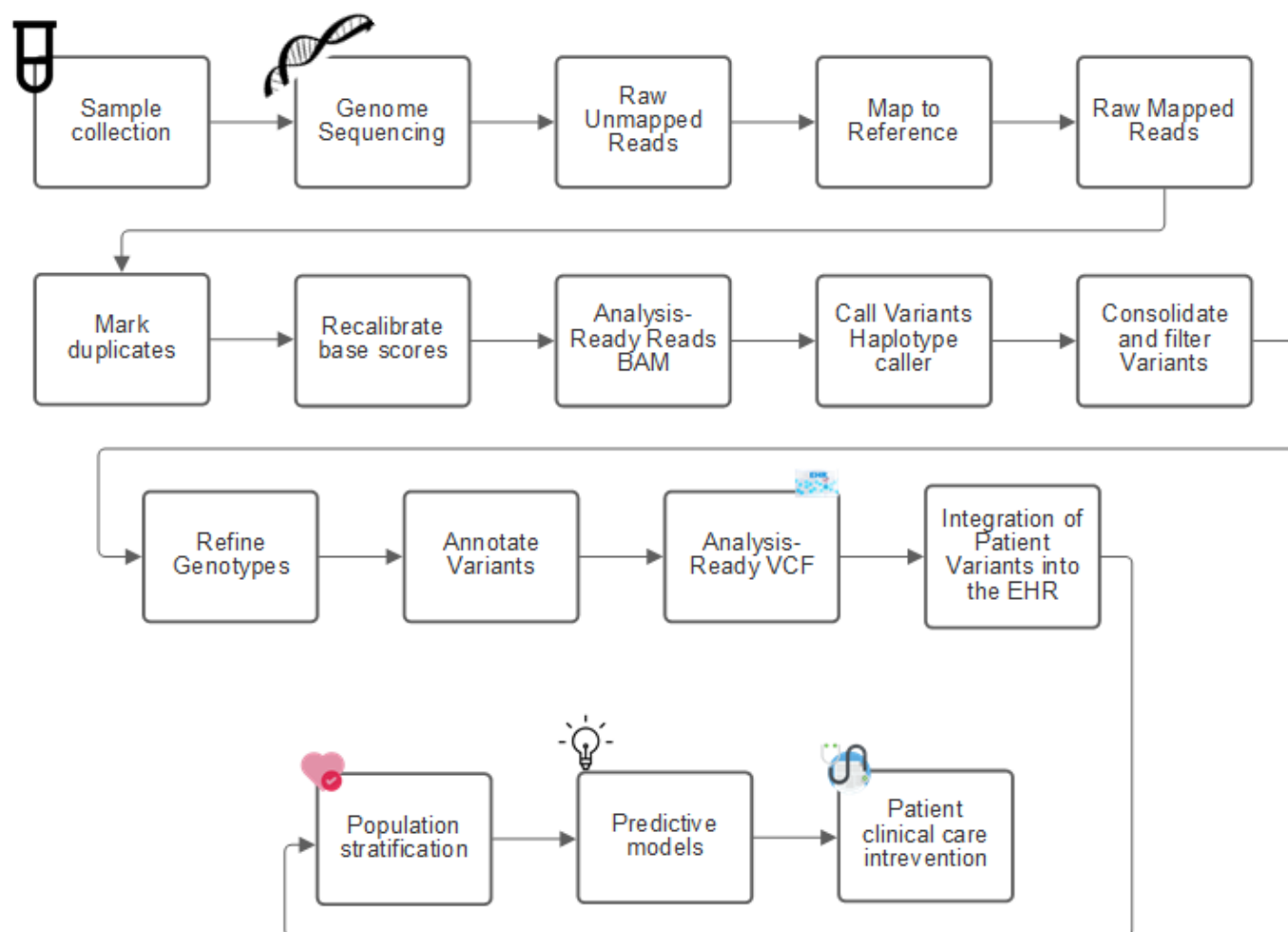
サンプルからバリエーションの識別、注釈、および予測まで

ゲノム解析の概要は、以下のステップに分類できます。これは完全なリストではありません。

1. サンプル収集。
2. **"ゲノム配列決定"** シーケンサーを使用して、raw データを生成します。
3. 前処理中です例：**"重複排除"** を使用します **"Picard"**。
4. ゲノム解析：
 - a. 参照ゲノムへのマッピング。
 - b. **"バリエーション"** GATK などのツールを使用して、一般的に識別とアノテーションを実行します。
5. 電子カルテ (EHR) システムへの統合
6. **"人口階層化"** 地理的位置と民族的背景を越えた遺伝的変動の同定。
7. **"予測モデル"** 有意なシングルヌクレオチド多形症を用いている。

8. "検証"。

次の図に、サンプリングからバリエントの識別、注釈、および予測までのプロセスを示します。



ヒトゲノム計画は 2003 年 4 月に完成し、このプロジェクトは、パブリックドメインで利用可能なヒトゲノム配列を非常に高品質でシミュレーションしました。このリファレンスゲノムは、ゲノム機能の研究開発で爆発的に増加しました。事実上すべての人間の病気にその人間の遺伝子の署名がある。最近まで、医師は、単一遺伝子の変化によって発生した特定の相続パターンによって引き起こされる、鎌状赤血球貧血などの出生異常を予測し、判定するために遺伝子を利用していた。ヒトゲノムプロジェクトで収集された膨大なデータがゲノム機能の最新状態に登場しました。

ゲノミクスには幅広いメリットがあります。ヘルスケアおよびライフサイエンス分野のメリットを以下に示します。

- 治療時点でのより良い診断
- 予後が良好である
- 精密医学
- パーソナライズされた治療計画
- 疾患モニタリングの向上
- 有害事象の減少

- 治療へのアクセスが向上しました
- 疾患モニタリングの改善
- 有効な臨床試験への参加と、遺伝子型に基づく臨床試験の患者の選択の向上。

ゲノミクスは a "4 つのヘッドを持つ獣、" 取得、ストレージ、分散、分析という、データセットのライフサイクル全体にわたるコンピューティングのニーズがあるためです。

ゲノム解析ツールキット（GATK）

GATK は、でデータサイエンスプラットフォームとして開発されました ["ブロードインスティテュート"](#)。GATK は、ゲノム解析を可能にする一連のオープンソース・ツールで、特に変異検出、同定、アノテーション、ジェノタイピングなどを行います。GATK の利点の 1 つは、ツールやコマンドのセットを連鎖させて、完全なワークフローを形成できることです。ブロード研究所が取り組む主な課題は、次のとおりです。

- 病気の根本原因と生物学的メカニズムを理解する。
- 疾患の基礎原因で作用する治療的介入を特定する。
- 変異体から人間の生理学的な機能まで、視線を理解します。
- 標準とポリシーを作成します ["フレームワーク"](#) ゲノムデータの表示、保存、分析、セキュリティなどを行います。
- 相互運用可能なゲノム集約データベース（gnomad）を標準化し、社会化します。
- ゲノムを用いたモニタリング、診断、および患者の治療をより正確に行うことができます。
- 症状が現れる前に疾患を適切に予測するツールの導入を支援します。
- 生物医学における最も困難で最も重要な問題に対処するために、学際的な協力者のコミュニティを作成し、強化します。

GATK と The Broad Institute によると、ゲノム配列決定は病理学ラボでプロトコルとして扱われるべきです。どのような作業でも、サンプルや実験全体でよく文書化され、最適化され、再現性があり、一貫性が保たれます。以下は、ブロード研究所が推奨する一連の手順です。詳細については、[を参照してください "GATK の Web サイト"](#)。

FlexPod セットアップ

ゲノミクスワークロードの検証には、FlexPod インフラプラットフォームのスクラッチからのセットアップが含まれています。FlexPod プラットフォームは高可用性を備えており、個別に拡張できます。たとえば、ネットワーク、ストレージ、コンピューティングを個別に拡張できます。FlexPod 環境をセットアップするためのリファレンスアーキテクチャドキュメントとして、次のシスコ検証済み設計ガイドを使用しました。 ["FlexPod Datacenter with VMware vSphere 7.0 and NetApp ONTAP 9.7" を参照してください](#)。次の FlexPod プラットフォームのセットアップのハイライトを参照してください。

FlexPod のラボセットアップを実行するには、次の手順を実行します。

1. FlexPod ラボのセットアップと検証では、次の IP4 予約と VLAN を使用します。

IP Reservations

VLAN	IP Range	Subnet Mask	Purpose
3281	172.21.25 /24	255.255.255.0	IB-MGMT
3282	172.21.26 /24	255.255.255.0	vMotion
3283	172.21.27 /24	255.255.255.0	VM
3284	172.21.28 /24	255.255.255.0	NFS
3285	172.21.29 /24	255.255.255.0	iSCSI-A
3286	172.21.30 /24	255.255.255.0	iSCSI-B

2. ONTAP SVM で iSCSI ベースのブート LUN を設定

ONTAP System Manager

LUNs

+ Add

	Name	Storage VM
▼	ESXi_Boot_Lun_1	Healthcare_SVM
▼	ESXi_Boot_Lun_2	Healthcare_SVM
▼	ESXi_Boot_Lun_3	Healthcare_SVM
▼	ESXi_Boot_Lun_4	Healthcare_SVM
▼	ESXi_Boot_Lun_5	Healthcare_SVM
▼	ESXi_Boot_Lun_6	Healthcare_SVM

3. LUN を iSCSI イニシエータグループにマッピングします。

	Name	Storage VM	Volume	Size	IOPS	Latency (ms)	Throughput (MB/s)
▲	ESXi_Boot_Lun_1	Healthcare_SVM	ESXi_Boot_Vol	20 GB	3	0.16	0.01

STATUS	VOLUME	DESCRIPTION	SNAPSHOT COPIES (LOCAL)	SNAPMIRROR (LOCAL OR REMOTE)
Online	ESXi_Boot_Vol	-	Protected	Unprotected

SERIAL NUMBER	QOS POLICY GROUP	MAPPED TO INITIATORS	ID	SNAPSHOT POLICY
80A4X+R8rAhP	-	GenomicsESXi_1 (1) iqn.1992-08.com.cisco:ucs-...	0	default

CAPACITY (AVAILABLE % TOTAL)	LUN FORMAT	PATH
95% 20 GB	VMware	/vol/ESXi_Boot_Vol/ESXi_Boot_Lun_1

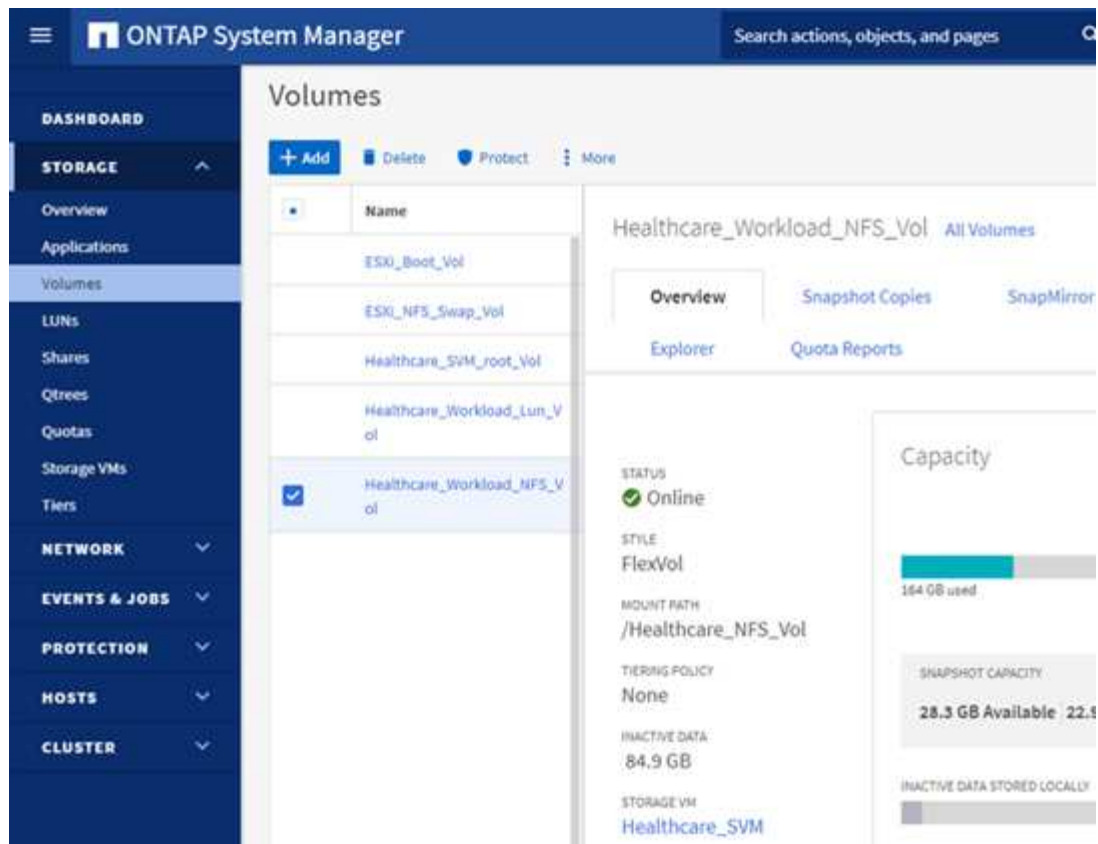
	Name	Storage VM	Volume	Size	IOPS	Latency (ms)	Throughput (MB/s)
▼	ESXi_Boot_Lun_1	Healthcare_SVM	ESXi_Boot_Vol	20 GB	1	0.25	0.01
▲	ESXi_Boot_Lun_2	Healthcare_SVM	ESXi_Boot_Vol	20 GB	4	0.18	0.02

STATUS Online	VOLUME ESXi_Boot_Vol	DESCRIPTION -	SNAPSHOT COPIES (LOCAL) STATUS Protected	SNAPMIRROR (LOCAL OR REMOTE) STATUS Unprotected
SERIAL NUMBER 80A4X+R8rAhU	QOS POLICY GROUP -	MAPPED TO INITIATORS GenomicsESXi_2 (1) iqn.1992-08.com.cisco:ucs-...	ID 0	SNAPSHOT POLICY default
CAPACITY (AVAILABLE % TOTAL) 96% 20 GB	LUN FORMAT VMware			

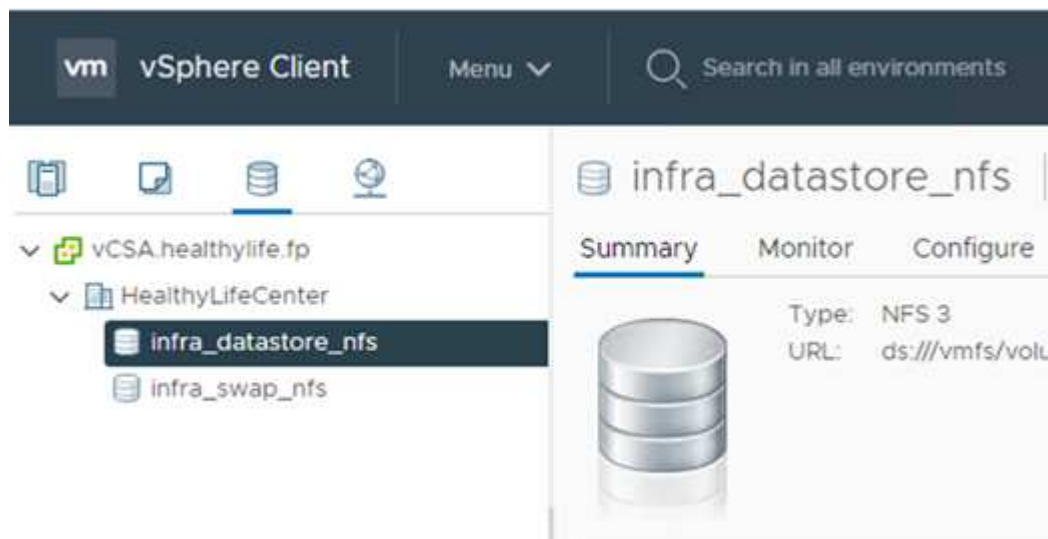
4. iSCSI ブートを使用して vSphere 7.0 をインストールします。
5. ESXi ホストを vCenter に登録します。



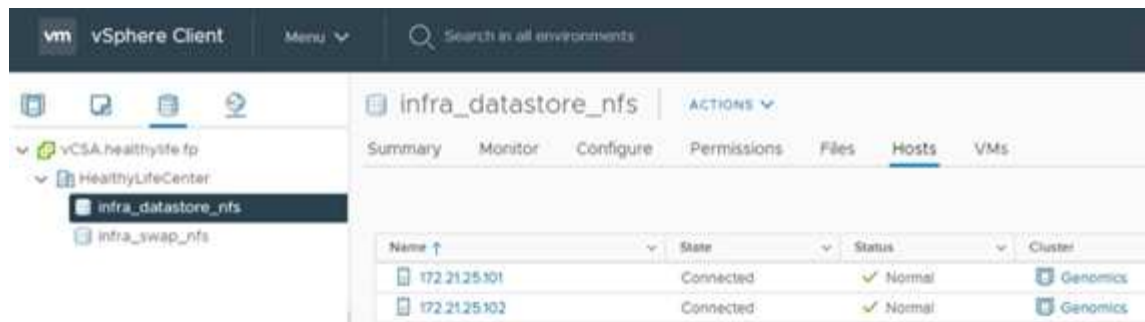
6. ONTAP ストレージ上で NFS データストア「infra_datastore_nfs」をプロビジョニングします。



7. vCenter にデータストアを追加します。



8. vCenter を使用して、ESXi ホストに NFS データストアを追加します。



9. vCenter を使用して、GATK を実行する Red Hat Enterprise Linux （RHEL） 8.3 VM を作成します。
10. NFS データストアが VM に提供され、「/mnt/ゲノミクス」でマウントされます。これは、GATK 実行可能ファイル、スクリプト、BAM（バイナリ・アライメント・マップ）ファイル、参照ファイル、インデックスファイル、辞書ファイル、およびバリエーション呼び出し用の出力ファイルを格納するために使用されます。

```
[root@genomics1 genomics]# df | grep genomics
/dev/sdb                308587328  5699492 287142812   2% /mnt/genomics
[root@genomics1 genomics]#
```

GATK のセットアップと実行

Red Hat Enterprise 8.3 Linux VM に次の前提条件をインストールします。

- Java 8 または SDK 1.8 以降
- Broad Institute から GATK 4.2.0.0 をダウンロードしてください ["GitHub サイト"](#)。一般に、ゲノム配列データは、タブ区切りの一連の ASCII カラムの形で保存されます。ただし、ASCII の保存に必要なスペースが多すぎます。したがって、新しい標準は BAM (*.bam) ファイルと呼ばれて進化しました。BAM ファイルは、シーケンスデータを圧縮、インデックス化、およびバイナリ形式で格納します。私たち ["ダウンロードしました"](#) から GATK を実行するために公開されている BAM ファイルのセット ["パブリックドメイン"](#)。インデックスファイル (*.bai)、辞書ファイル (*.dict)、および参照データファイル (*.fasta) を参照してください。

ダウンロード後、GATK ツールキットには jar ファイルと一連のサポートスクリプトがあります。

- GATK-PACKPACK-4.2.0.0-local.jar 実行可能ファイル
- 「GATK」スクリプトファイル。

父、母、息子 *。BAM ファイルで構成された家族の BAM ファイルと対応する索引、辞書、参照ゲノムファイルをダウンロードしました。

クロムウェルエンジン

Cromwell は、ワークフロー管理を可能にする科学的なワークフローを対象としたオープンソースエンジンです。クロムウェルエンジンは 2 つの方法で動作できます ["モード"](#)、サーバーモード、または単一ワークフローの実行モード。クロムウェルエンジンの動作は、を使用して制御できます ["クロムウェルエンジンコンフィギュレーションファイル"](#)。

- * サーバーモード。* 有効にします ["RESTful なホテル"](#) クロムウェルエンジンでのワークフローの実行。
- * 実行モード。* 実行モードはクロムウェルで単一のワークフローを実行する場合に最適です。 ["参照（](#)

[Ref](#)) " 実行モードで使用可能なすべてのオプションを表示します。

当社では、Cromwell エンジンを使用してワークフローとパイプラインを大規模に実行しています。クロムウェルエンジンは使いやすいエンジンです "[Workflow 概要の言語](#)" (WDL) ベースのスクリプト言語。また、Cromwell は、Common Workflow Language (CWL) と呼ばれる 2 つ目のワークフロースクリプト標準もサポートしています。このテクニカルレポートでは、WDL を使用しました。WDL は、もともと、広範なゲノム解析パイプライン研究所によって開発されたものです。WDL ワークフローを使用するには、次のようないくつかの戦略を使用します。

- * リニアチェーン。* 名前が示すように、タスク #1 からの出力がタスク #2 に入力として送信されます。
- * マルチイン / アウト。* これは、各タスクで複数の出力を後続のタスクに入力として送信できる点で、リニアチェーンと似ています。
- * Scatter-Gather * これは、特にイベント駆動型アーキテクチャで使用される場合に、最も強力なエンタープライズ・アプリケーション・インテグレーション (EAI) 戦略の 1 つです。各タスクは分離された方法で実行され、各タスクの出力が最終出力に統合されます。

WDL を使用してスタンドアロンモードで GATK を実行するには、次の 3 つの手順があります。

1. 「womtool.jar」を使用して構文を検証します。

```
[root@genomics1 ~]# java -jar womtool.jar validate ghplo.wdl
```

2. JSON の生成

```
[root@genomics1 ~]# java -jar womtool.jar inputs ghplo.wdl > ghplo.json
```

3. Cromwell エンジンと Cromwell.jar を使用してワークフローを実行します

```
[root@genomics1 ~]# java -jar cromwell.jar run ghplo.wdl --inputs ghplo.json
```

GATK は、いくつかのメソッドを使用して実行できます。このドキュメントでは、これらの方法のうちの 3 つについて説明します。

jar ファイルを使用した **GATK** の実行

では、hplotype バリエントの呼び出し側を使用した単一バリエントのコールパイプラインの実行について見てみましょう。

```
[root@genomics1 ~]# java -Dsamjdk.use_async_io_read_samtools=false \
-Dsamjdk.use_async_io_write_samtools=true \
-Dsamjdk.use_async_io_write_tribble=false \
-Dsamjdk.compression_level=2 \
-jar /mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-local.jar \
HaplotypeCaller \
--input /mnt/genomics/GATK/TEST\ DATA/bam/workshop_1906_2-
germline_bams_father.bam \
--output workshop_1906_2-germline_bams_father.validation.vcf \
--reference /mnt/genomics/GATK/TEST\ DATA/ref/workshop_1906_2-
germline_ref_ref.fasta
```

この実行方法では、GATK ローカル実行 jar ファイルを使用し、1つの Java コマンドを使用して jar ファイルを呼び出します。このコマンドには、いくつかのパラメータが渡されます。

1. このパラメータは 'HaplotypeCaller バリエントの呼び出し側パイプラインを呼び出していることを示します
2. '--input' は、入力 BAM ファイルを指定します。
3. '--output' は、variant 呼び出し形式 (*.VCF) でバリエント出力ファイルを指定します。([参照 \(Ref\)](#))
4. 「--reference」パラメータを使用して、参照ゲノムを渡しています。

実行すると、出力の詳細がセクションに表示されます ["jar ファイルを使用して GATK を実行するための出力。"](#)

./GATK スクリプトを使用した GATK の実行

GATK ツール・キットは './GATK' スクリプトを使用して実行できます次のコマンドを見てみましょう。

```
[root@genomics1 execution]# ./gatk \
--java-options "-Xmx4G" \
HaplotypeCaller \
-I /mnt/genomics/GATK/TEST\ DATA/bam/workshop_1906_2-
germline_bams_father.bam \
-R /mnt/genomics/GATK/TEST\ DATA/ref/workshop_1906_2-
germline_ref_ref.fasta \
-O /mnt/genomics/GATK/TEST\ DATA/variants.vcf
```

コマンドにはいくつかのパラメータを渡します。

- このパラメータは 'HaplotypeCaller バリエントの呼び出し側パイプラインを呼び出していることを示します
- 「-i」は、入力 BAM ファイルを指定します。
- 「-O」は、バリエント・コール・フォーマット (*.VCF) でバリエント出力ファイルを指定します。([参照 \(Ref\)](#))

- R パラメータを使用して、参照ゲノムを渡しています。

実行すると、出力の詳細がセクションに表示されます ["016e203cf9beada735f224ab14d0b3af"](#)

クロムウェルエンジンを使用した **GATK** の実行

当社では、クロムウェルエンジンを使用して GATK の実行を管理しています。コマンドラインとパラメータを見てみましょう。

```
[root@genomics1 genomics]# java -jar cromwell-65.jar \  
run /mnt/genomics/GATK/seq/ghplo.wdl \  
--inputs /mnt/genomics/GATK/seq/ghplo.json
```

ここでは '-car' パラメータを渡して java コマンドを呼び出しますこれは 'Cromwell-65.jar' などの jar ファイルを実行することを示します次に渡されるパラメータ ('run') は、クロムウェルエンジンが実行モードで実行されていることを示します。もう 1 つのオプションはサーバーモードです。次のパラメータは '*.wdl' ですこれは '実行モードがパイプラインを実行するために使用する必要があります次のパラメータは、実行するワークフローへの入力パラメータのセットです。

「ghplo.wdl」ファイルの内容は次のようになります。

```
[root@genomics1 seq]# cat ghplo.wdl  
workflow helloHaplotypeCaller {  
  call haplotypeCaller  
}  
task haplotypeCaller {  
  File GATK  
  File RefFasta  
  File RefIndex  
  File RefDict  
  String sampleName  
  File inputBAM  
  File bamIndex  
  command {  
    java -jar ${GATK} \  
      HaplotypeCaller \  
      -R ${RefFasta} \  
      -I ${inputBAM} \  
      -O ${sampleName}.raw.indels.snps.vcf  
  }  
  output {  
    File rawVCF = "${sampleName}.raw.indels.snps.vcf"  
  }  
}  
[root@genomics1 seq]#
```


ここでは、Cromwell エンジンへの入力を持つ、対応する JSON ファイルを示します。

```
[root@genomics1 seq]# cat ghplo.json
{
  "helloHaplotypeCaller.haplotypeCaller.GATK": "/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-local.jar",
  "helloHaplotypeCaller.haplotypeCaller.RefFasta": "/mnt/genomics/GATK/TEST DATA/ref/workshop_1906_2-germline_ref_ref.fasta",
  "helloHaplotypeCaller.haplotypeCaller.RefIndex": "/mnt/genomics/GATK/TEST DATA/ref/workshop_1906_2-germline_ref_ref.fasta.fai",
  "helloHaplotypeCaller.haplotypeCaller.RefDict": "/mnt/genomics/GATK/TEST DATA/ref/workshop_1906_2-germline_ref_ref.dict",
  "helloHaplotypeCaller.haplotypeCaller.sampleName": "fatherbam",
  "helloHaplotypeCaller.haplotypeCaller.inputBAM": "/mnt/genomics/GATK/TEST DATA/bam/workshop_1906_2-germline_bams_father.bam",
  "helloHaplotypeCaller.haplotypeCaller.bamIndex": "/mnt/genomics/GATK/TEST DATA/bam/workshop_1906_2-germline_bams_father.bai"
}
[root@genomics1 seq]#
```

Cromwell は実行にインメモリデータベースを使用していることに注意してください。実行すると、出力ログがセクションに表示されます ["クロムウェルエンジンを使用した GATK 実行用出力。"](#)

GATK を実行するための包括的な手順については、を参照してください ["GATK のドキュメント"](#)。

"次の例：jar ファイルを使用して GATK を実行するための出力。"

jar ファイルを使用して **GATK** を実行するための出力

"以前のゲノム - GATK のセットアップと実行。"

jar ファイルを使用して GATK を実行すると、次のような出力が得られます。

```
[root@genomics1 execution]# java -Dsamjdk.use_async_io_read_samtools=false \
-Dsamjdk.use_async_io_write_samtools=true \
-Dsamjdk.use_async_io_write_tribble=false \
-Dsamjdk.compression_level=2 \
-jar /mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-local.jar \
HaplotypeCaller \
--input /mnt/genomics/GATK/TEST\ DATA/bam/workshop_1906_2-germline_bams_father.bam \
--output workshop_1906_2-germline_bams_father.validation.vcf \
--reference /mnt/genomics/GATK/TEST\ DATA/ref/workshop_1906_2-germline_ref_ref.fasta \
```

```

22:52:58.430 INFO NativeLibraryLoader - Loading libgkl_compression.so
from jar:file:/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-
local.jar!/com/intel/gkl/native/libgkl_compression.so
Aug 17, 2021 10:52:58 PM
shaded.cloud_nio.com.google.auth.oauth2.ComputeEngineCredentials
runningOnComputeEngine
INFO: Failed to detect whether we are running on Google Compute Engine.
22:52:58.541 INFO HaplotypeCaller -
-----
22:52:58.542 INFO HaplotypeCaller - The Genome Analysis Toolkit (GATK)
v4.2.0.0
22:52:58.542 INFO HaplotypeCaller - For support and documentation go to
https://software.broadinstitute.org/gatk/
22:52:58.542 INFO HaplotypeCaller - Executing as
root@genomics1.healthyliife.fp on Linux v4.18.0-305.3.1.el8_4.x86_64 amd64
22:52:58.542 INFO HaplotypeCaller - Java runtime: OpenJDK 64-Bit Server
VM v1.8.0_302-b08
22:52:58.542 INFO HaplotypeCaller - Start Date/Time: August 17, 2021
10:52:58 PM EDT
22:52:58.542 INFO HaplotypeCaller -
-----
22:52:58.542 INFO HaplotypeCaller -
-----
22:52:58.542 INFO HaplotypeCaller - HTSJDK Version: 2.24.0
22:52:58.542 INFO HaplotypeCaller - Picard Version: 2.25.0
22:52:58.542 INFO HaplotypeCaller - Built for Spark Version: 2.4.5
22:52:58.542 INFO HaplotypeCaller - HTSJDK Defaults.COMPRESSION_LEVEL : 2
22:52:58.543 INFO HaplotypeCaller - HTSJDK
Defaults.USE_ASYNC_IO_READ_FOR_SAMTOOLS : false
22:52:58.543 INFO HaplotypeCaller - HTSJDK
Defaults.USE_ASYNC_IO_WRITE_FOR_SAMTOOLS : true
22:52:58.543 INFO HaplotypeCaller - HTSJDK
Defaults.USE_ASYNC_IO_WRITE_FOR_TRIBBLE : false
22:52:58.543 INFO HaplotypeCaller - Deflater: IntelDeflater
22:52:58.543 INFO HaplotypeCaller - Inflater: IntelInflater
22:52:58.543 INFO HaplotypeCaller - GCS max retries/reopens: 20
22:52:58.543 INFO HaplotypeCaller - Requester pays: disabled
22:52:58.543 INFO HaplotypeCaller - Initializing engine
22:52:58.804 INFO HaplotypeCaller - Done initializing engine
22:52:58.809 INFO HaplotypeCallerEngine - Disabling physical phasing,
which is supported only for reference-model confidence output
22:52:58.820 INFO NativeLibraryLoader - Loading libgkl_utils.so from
jar:file:/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-
local.jar!/com/intel/gkl/native/libgkl_utils.so
22:52:58.821 INFO NativeLibraryLoader - Loading libgkl_pairhmm_omp.so
from jar:file:/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-

```

```

local.jar!/com/intel/gkl/native/libgkl_pairhmm_omp.so
22:52:58.854 INFO   IntelPairHmm - Using CPU-supported AVX-512 instructions
22:52:58.854 INFO   IntelPairHmm - Flush-to-zero (FTZ) is enabled when
running PairHMM
22:52:58.854 INFO   IntelPairHmm - Available threads: 16
22:52:58.854 INFO   IntelPairHmm - Requested threads: 4
22:52:58.854 INFO   PairHMM - Using the OpenMP multi-threaded AVX-
accelerated native PairHMM implementation
22:52:58.872 INFO   ProgressMeter - Starting traversal
22:52:58.873 INFO   ProgressMeter -           Current Locus   Elapsed Minutes
Regions Processed   Regions/Minute
22:53:00.733 WARN   InbreedingCoeff - InbreedingCoeff will not be
calculated at position 20:9999900 and possibly subsequent; at least 10
samples must have called genotypes
22:53:08.873 INFO   ProgressMeter -           20:17538652           0.2
58900              353400.0
22:53:17.681 INFO   HaplotypeCaller - 405 read(s) filtered by:
MappingQualityReadFilter
0 read(s) filtered by: MappingQualityAvailableReadFilter
0 read(s) filtered by: MappedReadFilter
0 read(s) filtered by: NotSecondaryAlignmentReadFilter
6628 read(s) filtered by: NotDuplicateReadFilter
0 read(s) filtered by: PassesVendorQualityCheckReadFilter
0 read(s) filtered by: NonZeroReferenceLengthAlignmentReadFilter
0 read(s) filtered by: GoodCigarReadFilter
0 read(s) filtered by: WellformedReadFilter
7033 total reads filtered
22:53:17.681 INFO   ProgressMeter -           20:63024652           0.3
210522             671592.9
22:53:17.681 INFO   ProgressMeter - Traversal complete. Processed 210522
total regions in 0.3 minutes.
22:53:17.687 INFO   VectorLoglessPairHMM - Time spent in setup for JNI call
: 0.010347438
22:53:17.687 INFO   PairHMM - Total compute time in PairHMM
computeLogLikelihoods() : 0.259172573
22:53:17.687 INFO   SmithWatermanAligner - Total compute time in java
Smith-Waterman : 1.27 sec
22:53:17.687 INFO   HaplotypeCaller - Shutting down engine
[August 17, 2021 10:53:17 PM EDT]
org.broadinstitute.hellbender.tools.walkers.haplotypecaller.HaplotypeCalle
r done. Elapsed time: 0.32 minutes.
Runtime.totalMemory()=5561122816
[root@genomics1 execution]#

```

出力ファイルは、実行後に指定された場所にあります。

./GATK スクリプトを使用して GATK を実行するための出力

"Previous : jar ファイルを使用して GATK を実行するための出力。"

「./GATK」スクリプトを使用して GATK を実行すると、次の出力例が得られます。

```
[root@genomics1 gatk-4.2.0.0]# ./gatk --java-options "-Xmx4G" \
HaplotypeCaller \
-I /mnt/genomics/GATK/TEST\ DATA/bam/workshop_1906_2-
germline_bams_father.bam \
-R /mnt/genomics/GATK/TEST\ DATA/ref/workshop_1906_2-
germline_ref_ref.fasta \
-O /mnt/genomics/GATK/TEST\ DATA/variants.vcf
Using GATK jar /mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-
local.jar
Running:
    java -Dsamjdk.use_async_io_read_samtools=false
-Dsamjdk.use_async_io_write_samtools=true
-Dsamjdk.use_async_io_write_tribble=false -Dsamjdk.compression_level=2
-Xmx4G -jar /mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-local.jar
HaplotypeCaller -I /mnt/genomics/GATK/TEST DATA/bam/workshop_1906_2-
germline_bams_father.bam -R /mnt/genomics/GATK/TEST
DATA/ref/workshop_1906_2-germline_ref_ref.fasta -O /mnt/genomics/GATK/TEST
DATA/variants.vcf
23:29:45.553 INFO  NativeLibraryLoader - Loading libgkl_compression.so
from jar:file:/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-
local.jar!/com/intel/gkl/native/libgkl_compression.so
Aug 17, 2021 11:29:45 PM
shaded.cloud_nio.com.google.auth.oauth2.ComputeEngineCredentials
runningOnComputeEngine
INFO: Failed to detect whether we are running on Google Compute Engine.
23:29:45.686 INFO  HaplotypeCaller -
-----
23:29:45.686 INFO  HaplotypeCaller - The Genome Analysis Toolkit (GATK)
v4.2.0.0
23:29:45.686 INFO  HaplotypeCaller - For support and documentation go to
https://software.broadinstitute.org/gatk/
23:29:45.687 INFO  HaplotypeCaller - Executing as
root@genomics1.healthyliife.fp on Linux v4.18.0-305.3.1.el8_4.x86_64 amd64
23:29:45.687 INFO  HaplotypeCaller - Java runtime: OpenJDK 64-Bit Server
VM v11.0.12+7-LTS
23:29:45.687 INFO  HaplotypeCaller - Start Date/Time: August 17, 2021 at
11:29:45 PM EDT
23:29:45.687 INFO  HaplotypeCaller -
```

```

-----
23:29:45.687 INFO HaplotypeCaller -
-----
23:29:45.687 INFO HaplotypeCaller - HTSJDK Version: 2.24.0
23:29:45.687 INFO HaplotypeCaller - Picard Version: 2.25.0
23:29:45.687 INFO HaplotypeCaller - Built for Spark Version: 2.4.5
23:29:45.688 INFO HaplotypeCaller - HTSJDK Defaults.COMPRESSION_LEVEL : 2
23:29:45.688 INFO HaplotypeCaller - HTSJDK
Defaults.USE_ASYNC_IO_READ_FOR_SAMTOOLS : false
23:29:45.688 INFO HaplotypeCaller - HTSJDK
Defaults.USE_ASYNC_IO_WRITE_FOR_SAMTOOLS : true
23:29:45.688 INFO HaplotypeCaller - HTSJDK
Defaults.USE_ASYNC_IO_WRITE_FOR_TRIBBLE : false
23:29:45.688 INFO HaplotypeCaller - Deflater: IntelDeflater
23:29:45.688 INFO HaplotypeCaller - Inflater: IntelInflater
23:29:45.688 INFO HaplotypeCaller - GCS max retries/reopens: 20
23:29:45.688 INFO HaplotypeCaller - Requester pays: disabled
23:29:45.688 INFO HaplotypeCaller - Initializing engine
23:29:45.804 INFO HaplotypeCaller - Done initializing engine
23:29:45.809 INFO HaplotypeCallerEngine - Disabling physical phasing,
which is supported only for reference-model confidence output
23:29:45.818 INFO NativeLibraryLoader - Loading libgkl_utils.so from
jar:file:/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-
local.jar!/com/intel/gkl/native/libgkl_utils.so
23:29:45.819 INFO NativeLibraryLoader - Loading libgkl_pairhmm_omp.so
from jar:file:/mnt/genomics/GATK/gatk-4.2.0.0/gatk-package-4.2.0.0-
local.jar!/com/intel/gkl/native/libgkl_pairhmm_omp.so
23:29:45.852 INFO IntelPairHmm - Using CPU-supported AVX-512 instructions
23:29:45.852 INFO IntelPairHmm - Flush-to-zero (FTZ) is enabled when
running PairHMM
23:29:45.852 INFO IntelPairHmm - Available threads: 16
23:29:45.852 INFO IntelPairHmm - Requested threads: 4
23:29:45.852 INFO PairHMM - Using the OpenMP multi-threaded AVX-
accelerated native PairHMM implementation
23:29:45.868 INFO ProgressMeter - Starting traversal
23:29:45.868 INFO ProgressMeter -          Current Locus  Elapsed Minutes
Regions Processed  Regions/Minute
23:29:47.772 WARN InbreedingCoeff - InbreedingCoeff will not be
calculated at position 20:9999900 and possibly subsequent; at least 10
samples must have called genotypes
23:29:55.868 INFO ProgressMeter -          20:18885652          0.2
63390          380340.0
23:30:04.389 INFO HaplotypeCaller - 405 read(s) filtered by:
MappingQualityReadFilter
0 read(s) filtered by: MappingQualityAvailableReadFilter
0 read(s) filtered by: MappedReadFilter

```



```

0 read(s) filtered by: NotSecondaryAlignmentReadFilter
6628 read(s) filtered by: NotDuplicateReadFilter
0 read(s) filtered by: PassesVendorQualityCheckReadFilter
0 read(s) filtered by: NonZeroReferenceLengthAlignmentReadFilter
0 read(s) filtered by: GoodCigarReadFilter
0 read(s) filtered by: WellformedReadFilter
7033 total reads filtered
23:30:04.389 INFO ProgressMeter - 20:63024652 0.3
210522 681999.9
23:30:04.389 INFO ProgressMeter - Traversal complete. Processed 210522
total regions in 0.3 minutes.
23:30:04.395 INFO VectorLoglessPairHMM - Time spent in setup for JNI call
: 0.0121292030000000002
23:30:04.395 INFO PairHMM - Total compute time in PairHMM
computeLogLikelihoods() : 0.267345217
23:30:04.395 INFO SmithWatermanAligner - Total compute time in java
Smith-Waterman : 1.23 sec
23:30:04.395 INFO HaplotypeCaller - Shutting down engine
[August 17, 2021 at 11:30:04 PM EDT]
org.broadinstitute.hellbender.tools.walkers.haplotypecaller.HaplotypeCalle
r done. Elapsed time: 0.31 minutes.
Runtime.totalMemory()=2111832064
[root@genomics1 gatk-4.2.0.0]#

```

出力ファイルは、実行後に指定された場所にあります。

"次に、クロムウェルエンジンを使用した GATK の実行出力を示します。"

クロムウェルエンジンを使用した **GATK** 実行用出力

"11fffe01d469840980d9b9a5f45bf9ed"

Cromwell エンジンを使用して GATK を実行すると、次の出力例が得られます。

```

[root@genomics1 genomics]# java -jar cromwell-65.jar run
/mnt/genomics/GATK/seq/ghplo.wdl --inputs
/mnt/genomics/GATK/seq/ghplo.json
[2021-08-18 17:10:50,78] [info] Running with database db.url =
jdbc:hsqldb:mem:856a1f0d-9a0d-42e5-9199-
5e6c1d0f72dd;shutdown=false;hsqldb.tx=mvcc
[2021-08-18 17:10:57,74] [info] Running migration
RenameWorkflowOptionsInMetadata with a read batch size of 100000 and a
write batch size of 100000
[2021-08-18 17:10:57,75] [info] [RenameWorkflowOptionsInMetadata] 100%
[2021-08-18 17:10:57,83] [info] Running with database db.url =

```

```

jdbc:hsqldb:mem:6afe0252-2dc9-4e57-8674-
ce63c67aa142;shutdown=false;hsqldb.tx=mvcc
[2021-08-18 17:10:58,17] [info] Slf4jLogger started
[2021-08-18 17:10:58,33] [info] Workflow heartbeat configuration:
{
  "cromwellId" : "cromid-41b7e30",
  "heartbeatInterval" : "2 minutes",
  "ttl" : "10 minutes",
  "failureShutdownDuration" : "5 minutes",
  "writeBatchSize" : 10000,
  "writeThreshold" : 10000
}
[2021-08-18 17:10:58,38] [info] Metadata summary refreshing every 1
second.
[2021-08-18 17:10:58,38] [info] No metadata archiver defined in config
[2021-08-18 17:10:58,38] [info] No metadata deleter defined in config
[2021-08-18 17:10:58,40] [info] KvWriteActor configured to flush with
batch size 200 and process rate 5 seconds.
[2021-08-18 17:10:58,40] [info] WriteMetadataActor configured to flush
with batch size 200 and process rate 5 seconds.
[2021-08-18 17:10:58,44] [info] CallCacheWriteActor configured to flush
with batch size 100 and process rate 3 seconds.
[2021-08-18 17:10:58,44] [warn] 'docker.hash-lookup.gcr-api-queries-per-
100-seconds' is being deprecated, use 'docker.hash-lookup.gcr.throttle'
instead (see reference.conf)
[2021-08-18 17:10:58,54] [info] JobExecutionTokenDispenser - Distribution
rate: 50 per 1 seconds.
[2021-08-18 17:10:58,58] [info] SingleWorkflowRunnerActor: Version 65
[2021-08-18 17:10:58,58] [info] SingleWorkflowRunnerActor: Submitting
workflow
[2021-08-18 17:10:58,64] [info] Unspecified type (Unspecified version)
workflow 3e246147-b1a9-41dc-8679-319f81b7701e submitted
[2021-08-18 17:10:58,66] [info] SingleWorkflowRunnerActor: Workflow
submitted 3e246147-b1a9-41dc-8679-319f81b7701e
[2021-08-18 17:10:58,66] [info] 1 new workflows fetched by cromid-41b7e30:
3e246147-b1a9-41dc-8679-319f81b7701e
[2021-08-18 17:10:58,67] [info] WorkflowManagerActor: Starting workflow
3e246147-b1a9-41dc-8679-319f81b7701e
[2021-08-18 17:10:58,68] [info] WorkflowManagerActor: Successfully started
WorkflowActor-3e246147-b1a9-41dc-8679-319f81b7701e
[2021-08-18 17:10:58,68] [info] Retrieved 1 workflows from the
WorkflowStoreActor
[2021-08-18 17:10:58,70] [info] WorkflowStoreHeartbeatWriteActor
configured to flush with batch size 10000 and process rate 2 minutes.
[2021-08-18 17:10:58,76] [info] MaterializeWorkflowDescriptorActor
[3e246147]: Parsing workflow as WDL draft-2

```

```

[2021-08-18 17:10:59,34] [info] MaterializeWorkflowDescriptorActor
[3e246147]: Call-to-Backend assignments:
helloHaplotypeCaller.haplotypeCaller -> Local
[2021-08-18 17:11:00,54] [info] WorkflowExecutionActor-3e246147-b1a9-41dc-
8679-319f81b7701e [3e246147]: Starting
helloHaplotypeCaller.haplotypeCaller
[2021-08-18 17:11:01,56] [info] Assigned new job execution tokens to the
following groups: 3e246147: 1
[2021-08-18 17:11:01,70] [info] BackgroundConfigAsyncJobExecutionActor
[3e246147helloHaplotypeCaller.haplotypeCaller:NA:1]: java -jar
/mnt/genomics/cromwell-executions/helloHaplotypeCaller/3e246147-b1a9-41dc-
8679-319f81b7701e/call-haplotypeCaller/inputs/-179397211/gatk-package-
4.2.0.0-local.jar \
    HaplotypeCaller \
    -R /mnt/genomics/cromwell-executions/helloHaplotypeCaller/3e246147-
b1a9-41dc-8679-319f81b7701e/call-
haplotypeCaller/inputs/604632695/workshop_1906_2-germline_ref_ref.fasta \
    -I /mnt/genomics/cromwell-executions/helloHaplotypeCaller/3e246147-
b1a9-41dc-8679-319f81b7701e/call-
haplotypeCaller/inputs/604617202/workshop_1906_2-germline_bams_father.bam
\
    -O fatherbam.raw.indels.snps.vcf
[2021-08-18 17:11:01,72] [info] BackgroundConfigAsyncJobExecutionActor
[3e246147helloHaplotypeCaller.haplotypeCaller:NA:1]: executing: /bin/bash
/mnt/genomics/cromwell-executions/helloHaplotypeCaller/3e246147-b1a9-41dc-
8679-319f81b7701e/call-haplotypeCaller/execution/script
[2021-08-18 17:11:03,49] [info] BackgroundConfigAsyncJobExecutionActor
[3e246147helloHaplotypeCaller.haplotypeCaller:NA:1]: job id: 26867
[2021-08-18 17:11:03,53] [info] BackgroundConfigAsyncJobExecutionActor
[3e246147helloHaplotypeCaller.haplotypeCaller:NA:1]: Status change from -
to WaitingForReturnCode
[2021-08-18 17:11:03,54] [info] Not triggering log of token queue status.
Effective log interval = None
[2021-08-18 17:11:23,65] [info] BackgroundConfigAsyncJobExecutionActor
[3e246147helloHaplotypeCaller.haplotypeCaller:NA:1]: Status change from
WaitingForReturnCode to Done
[2021-08-18 17:11:25,04] [info] WorkflowExecutionActor-3e246147-b1a9-41dc-
8679-319f81b7701e [3e246147]: Workflow helloHaplotypeCaller complete.
Final Outputs:
{
    "helloHaplotypeCaller.haplotypeCaller.rawVCF": "/mnt/genomics/cromwell-
executions/helloHaplotypeCaller/3e246147-b1a9-41dc-8679-319f81b7701e/call-
haplotypeCaller/execution/fatherbam.raw.indels.snps.vcf"
}
[2021-08-18 17:11:28,43] [info] WorkflowManagerActor: Workflow actor for
3e246147-b1a9-41dc-8679-319f81b7701e completed with status 'Succeeded'.

```

The workflow will be removed from the workflow store.

[2021-08-18 17:11:32,24] [info] SingleWorkflowRunnerActor workflow finished with status 'Succeeded'.

```
{
  "outputs": {
    "helloHaplotypeCaller.haplotypeCaller.rawVCF":
"/mnt/genomics/cromwell-executions/helloHaplotypeCaller/3e246147-b1a9-41dc-8679-319f81b7701e/call-haplotypeCaller/execution/fatherbam.raw.indels.snps.vcf"
  },
  "id": "3e246147-b1a9-41dc-8679-319f81b7701e"
}
```

[2021-08-18 17:11:33,45] [info] Workflow polling stopped

[2021-08-18 17:11:33,46] [info] 0 workflows released by cromid-41b7e30

[2021-08-18 17:11:33,46] [info] Shutting down WorkflowStoreActor - Timeout = 5 seconds

[2021-08-18 17:11:33,46] [info] Shutting down WorkflowLogCopyRouter - Timeout = 5 seconds

[2021-08-18 17:11:33,46] [info] Shutting down JobExecutionTokenDispenser - Timeout = 5 seconds

[2021-08-18 17:11:33,46] [info] Aborting all running workflows.

[2021-08-18 17:11:33,46] [info] JobExecutionTokenDispenser stopped

[2021-08-18 17:11:33,46] [info] WorkflowStoreActor stopped

[2021-08-18 17:11:33,47] [info] WorkflowLogCopyRouter stopped

[2021-08-18 17:11:33,47] [info] Shutting down WorkflowManagerActor - Timeout = 3600 seconds

[2021-08-18 17:11:33,47] [info] WorkflowManagerActor: All workflows finished

[2021-08-18 17:11:33,47] [info] WorkflowManagerActor stopped

[2021-08-18 17:11:33,64] [info] Connection pools shut down

[2021-08-18 17:11:33,64] [info] Shutting down SubWorkflowStoreActor - Timeout = 1800 seconds

[2021-08-18 17:11:33,64] [info] Shutting down JobStoreActor - Timeout = 1800 seconds

[2021-08-18 17:11:33,64] [info] Shutting down CallCacheWriteActor - Timeout = 1800 seconds

[2021-08-18 17:11:33,64] [info] SubWorkflowStoreActor stopped

[2021-08-18 17:11:33,64] [info] Shutting down ServiceRegistryActor - Timeout = 1800 seconds

[2021-08-18 17:11:33,64] [info] Shutting down DockerHashActor - Timeout = 1800 seconds

[2021-08-18 17:11:33,64] [info] Shutting down IoProxy - Timeout = 1800 seconds

[2021-08-18 17:11:33,64] [info] CallCacheWriteActor Shutting down: 0 queued messages to process

[2021-08-18 17:11:33,64] [info] JobStoreActor stopped

```
[2021-08-18 17:11:33,64] [info] CallCacheWriteActor stopped
[2021-08-18 17:11:33,64] [info] KvWriteActor Shutting down: 0 queued
messages to process
[2021-08-18 17:11:33,64] [info] IoProxy stopped
[2021-08-18 17:11:33,64] [info] WriteMetadataActor Shutting down: 0 queued
messages to process
[2021-08-18 17:11:33,65] [info] ServiceRegistryActor stopped
[2021-08-18 17:11:33,65] [info] DockerHashActor stopped
[2021-08-18 17:11:33,67] [info] Database closed
[2021-08-18 17:11:33,67] [info] Stream materializer shut down
[2021-08-18 17:11:33,67] [info] WDL HTTP import resolver closed
[root@genomics1 genomics]#
```

"次は GPU のセットアップです。"

GPU セットアップ

"以前：クロムウェルエンジンを使用した GATK の実行出力。"

GATK ツールは、公開時点で、オンプレミスでの GPU ベースの実行をネイティブでサポートしていません。以下のセットアップとガイダンスは、GATK 用の PCIe メザニンカードを使用して、背面取り付けの NVIDIA Tesla P6 GPU で FlexPod を使用する場合の簡単な方法を読者が理解できるようにします。

次の Cisco Validated Design (CVD) をリファレンスアーキテクチャとして使用し、FlexPod 環境をセットアップして GPU を使用するアプリケーションを実行できるようにしました。

- ["FlexPod Datacenter for AI / ML with Cisco UCS 480 ML for Deep Learning"』を参照してください](#)

このセットアップの重要なポイントを次に示します。

1. UCS B200 M5 サーバのメザニンスロットに PCIe NVIDIA Tesla P6 GPU を使用しました。

Equipment / Chassis / Chassis 1 / Servers / Server 1

< General Inventory Virtual Machines Installed Firmware CIMC Sessions SEL Logs VIF Paths Health >

< Motherboard CIMC CPUs GPUs Memory Adapters HBAs NICs iSCSI vNICs Security >

▼ Advanced Filter

↑ Export

🖨 Print

⚙

Name	ID	Model	Serial	Mode
Graphics Card 2	2	UCSB-GPU-P6-R	FCH212373V7	Compute

Equipment / Chassis / Chassis 1 / Servers / Server 2

<GeneralInventoryVirtual MachinesInstalled FirmwareCIMC SessionsSEL LogsVIF PathsHealth>

<MotherboardCIMCCPUGPUsMemoryAdaptersHBAsNICsISCSI vNICsSecurity>

Advanced Filter

Export

Print

Name	ID	Model	Serial	Mode
Graphics Card 2	2	UCSB-GPU-P6-R	FCH212373Y1	Compute

- このセットアップでは、NVIDIA パートナーポータルに登録し、コンピューティングモードで GPU を使用できる評価用ライセンス（使用権）を取得しました。
- NVIDIA パートナーの Web サイトから、必要な NVIDIA vGPU ソフトウェアをダウンロードしました。
- エンタイトルメント「*.bin」ファイルを NVIDIA パートナーの Web サイトからダウンロードしました。
- NVIDIA vGPU ライセンスサーバをインストールし、NVIDIA パートナーサイトからダウンロードした「*.bin」ファイルを使用してライセンスサーバに使用権を追加しました。
- NVIDIA パートナーポータルで、導入環境に適した NVIDIA vGPU ソフトウェアのバージョンを選択してください。このセットアップでは、ドライバのバージョン 460.73.02 を使用しました。
- このコマンドは、をインストールします ["NVIDIA vGPU Manager の略"](#) ESXi で。

```
[root@localhost:~] esxcli software vib install -v
/vmfs/volumes/infra_datastore_nfs/nvidia/vib/NVIDIA_bootbank_NVIDIA-
VMware_ESXi_7.0_Host_Driver_460.73.02-10EM.700.0.0.15525992.vib
Installation Result
Message: Operation finished successfully.
Reboot Required: false
VIBs Installed: NVIDIA_bootbank_NVIDIA-
VMware_ESXi_7.0_Host_Driver_460.73.02-10EM.700.0.0.15525992
VIBs Removed:
VIBs Skipped:
```

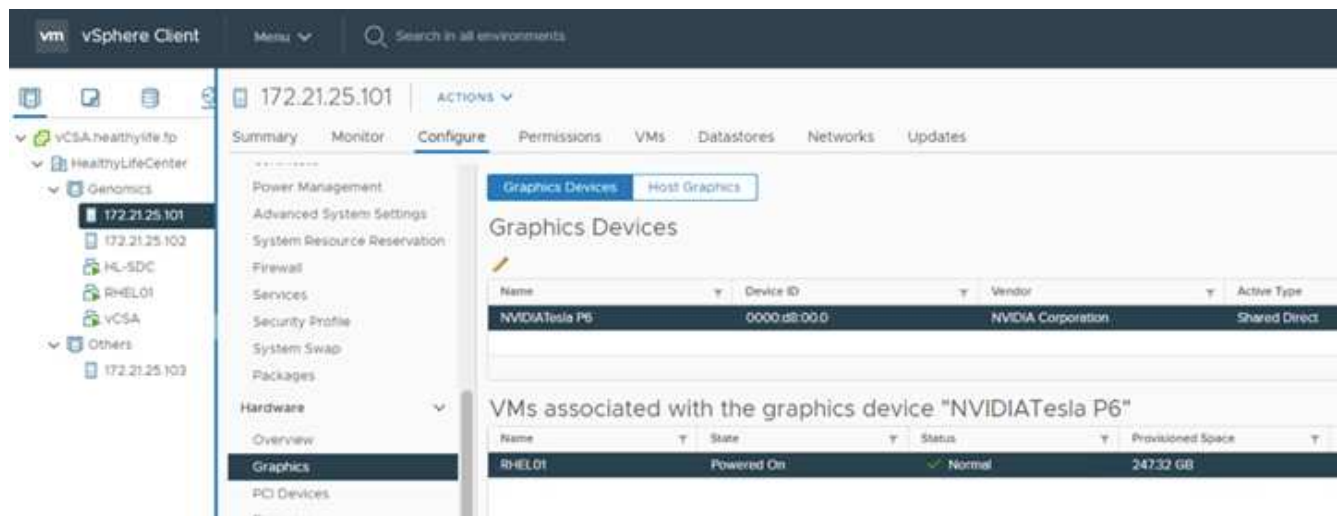
- ESXi サーバのリブート後、次のコマンドを実行してインストールを検証し、GPU の健全性を確認します。


```

[root@localhost:~] nvidia-smi
Wed Aug 18 21:37:19 2021
+-----+
+-----+
| NVIDIA-SMI 460.73.02      Driver Version: 460.73.02      CUDA Version: N/A
|
|-----+-----+
+-----+
| GPU   Name               Persistence-M| Bus-Id        Disp.A | Volatile
Uncorr. ECC |
| Fan   Temp   Perf   Pwr:Usage/Cap|      Memory-Usage | GPU-Util
Compute M. |
|
|-----+-----+
MIG M. |
|=====+=====+=====+
=====|
|    0   Tesla P6             On      | 00000000:D8:00.0 Off |
0 |
| N/A    35C    P8      9W /  90W |  15208MiB / 15359MiB |      0%
Default |
|
|-----+-----+
N/A |
+-----+-----+
+-----+
+-----+
+-----+
+-----+
| Processes:
|
| GPU   GI    CI          PID    Type    Process name          GPU
Memory |
|      ID    ID                 |          Usage
|
|=====+=====+=====+
=====|
|    0   N/A   N/A   2812553      C+G     RHEL01
15168MiB |
+-----+-----+
+-----+
[root@localhost:~]

```

9. vCenter を使用 "設定" グラフィックデバイスの設定は「Shared Direct」になります。



10. RedHat VM のセキュアブートが無効になっていることを確認します。
11. VM 起動オプションファームウェアが EFI ("参照 (Ref) ") 。

Edit Settings
RHEL01

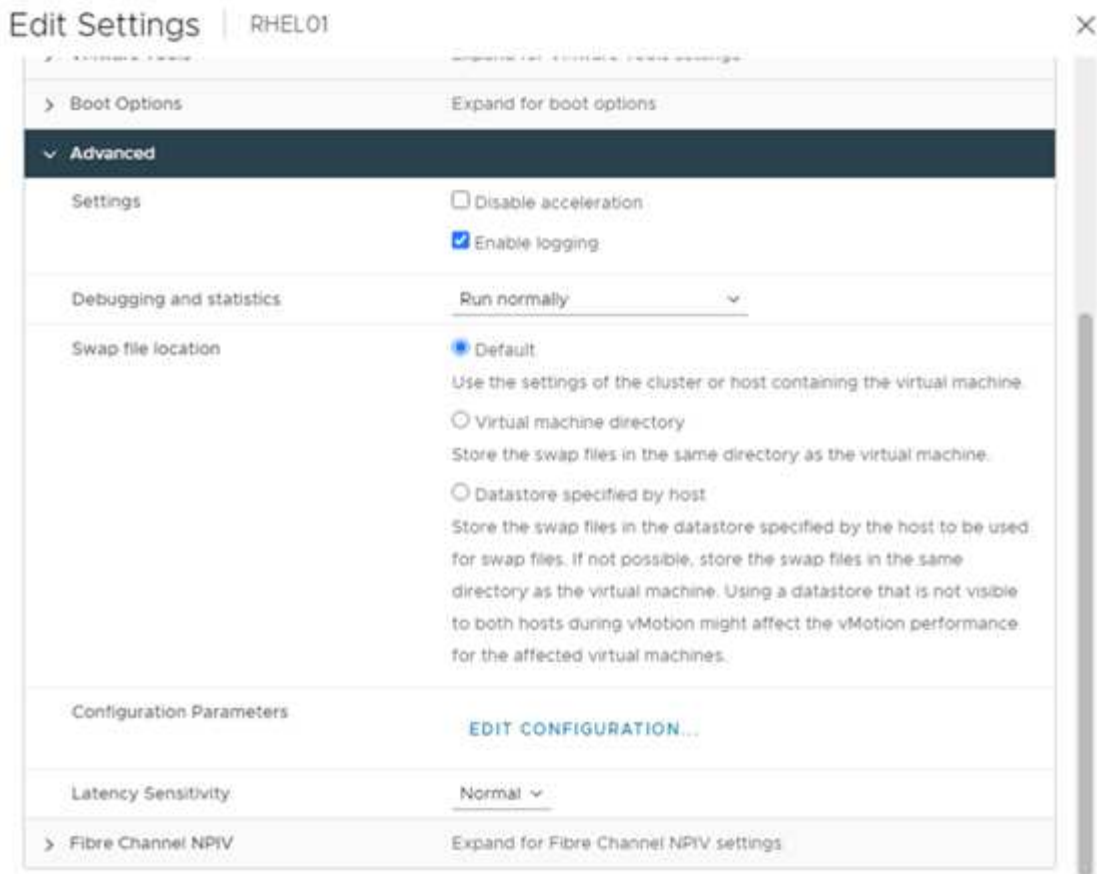
Virtual Hardware
VM Options

> General Options	VM Name: RHEL01
> VMware Remote Console Options	<input type="checkbox"/> Lock the guest operating system when the last remote user disconnects
> Encryption	Expand for encryption settings
> Power management	Expand for power management settings
> VMware Tools	Expand for VMware Tools settings
> Boot Options	
Firmware	EFI (recommended) ▼
Secure Boot	<input type="checkbox"/> Enabled
Boot Delay	When powering on or resetting, delay boot order by 0 milliseconds
Force EFI setup	<input type="checkbox"/> During the next boot, force entry into the EFI setup screen
Failed Boot Recovery	<input type="checkbox"/> If the VM fails to find boot device, automatically retry after 10 seconds
> Advanced	Expand for advanced settings
> Fibre Channel NPIV	Expand for Fibre Channel NPIV settings

CANCEL
OK

12. 次のパラメータが VM オプションの詳細編集設定に追加されていることを確認します。「pciPassthru.64bitMMIOSizeGB」パラメータの値は、VM に割り当てられた GPU のメモリと数によって異なります。例：

- VM に 32GB V100 GPU が 4 つ割り当てられている場合は、この値を 128 にします。
- VM に 16GB P6 GPU が 4 つ割り当てられている場合、この値は 64 である必要があります。

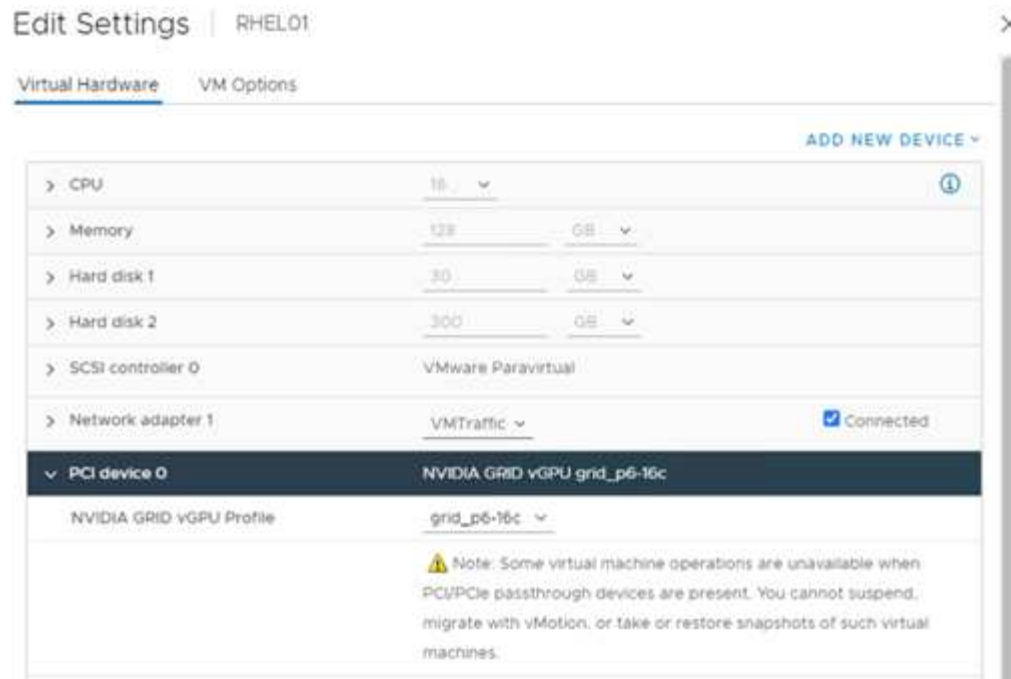


Configuration Parameters ✕

 Modify or add configuration parameters as needed for experimental features or as instructed by technical support. Empty values will be removed (supported on ESXi 6.0 and later).

Name 	Value 
pciPassthru.64bitMMIOSizeGB	64
pciPassthru.use64bitMMIO	TRUE

13. vCenter で新しい PCI デバイスとして vGPU を仮想マシンに追加する場合は、PCI デバイスタイプとして NVIDIA GRID vGPU を選択してください。
14. 使用している GPU、GPU メモリ、および使用目的を調整する適切な GPU プロファイルを選択します。たとえば、グラフィックスとコンピューティングです。



15. Red Hat Linux VM で、次のコマンドを実行して NVIDIA ドライバをインストールできます。

```
[root@genomics1 genomics]# sh NVIDIA-Linux-x86_64-460.73.01-grid.run
```

16. 次のコマンドを実行して、正しい vGPU プロファイルが報告されていることを確認します。

```
[root@genomics1 genomics]# nvidia-smi -query-gpu=gpu_name  
-format=csv,noheader -id=0 | sed -e 's/ /-/g'  
GRID-P6-16C  
[root@genomics1 genomics]#
```

17. リブート後は、正しい NVIDIA vGPU がドライバのバージョンと一緒に報告されていることを確認します。

```

[root@genomics1 genomics]# nvidia-smi
Wed Aug 18 20:30:56 2021
+-----+
+-----+
| NVIDIA-SMI 460.73.01      Driver Version: 460.73.01      CUDA Version:
11.2      |
|-----+-----+
+-----+
| GPU  Name           Persistence-M| Bus-Id        Disp.A | Volatile
Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util
Compute M. |
|
| MIG M. |
|=====+=====+=====
=====|
|   0  GRID P6-16C           On   | 00000000:02:02.0 Off |
N/A |
| N/A   N/A    P8    N/A /  N/A |   2205MiB / 16384MiB |      0%
Default |
|
|
N/A |
+-----+-----+
+-----+
+-----+
+-----+
+-----+
| Processes:
|
| GPU    GI    CI          PID    Type    Process name                  GPU
Memory |
|          ID    ID                                   Usage
|
|=====+=====+=====
=====|
|   0    N/A  N/A        8604      G    /usr/libexec/Xorg
13MiB |
+-----+-----+
+-----+
[root@genomics1 genomics]#

```

18. vGPU グリッド構成ファイルの VM にライセンスサーバの IP が設定されていることを確認してください。

a. テンプレートをコピーします。

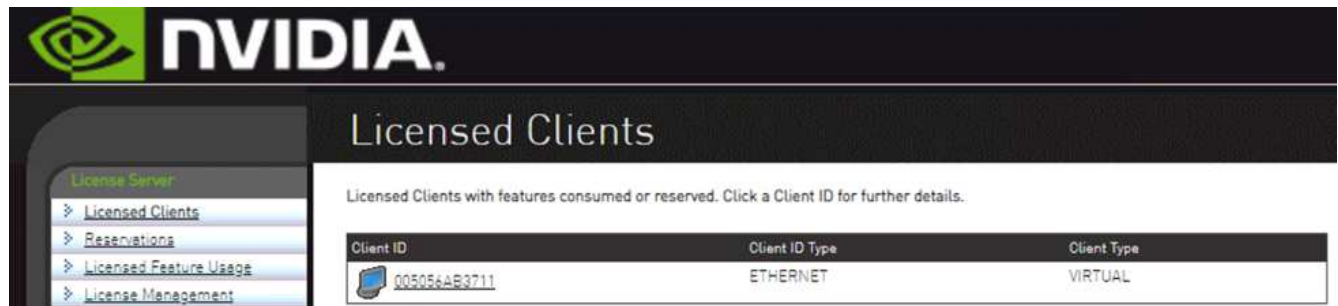

```
[root@genomics1 genomics]# cp /etc/nvidia/gridd.conf.template  
/etc/nvidia/gridd.conf
```

- b. /etc/nvidia /rid ファイルを編集し、ライセンス・サーバの IP アドレスを追加して、機能タイプを 1 に設定します。

```
ServerAddress=192.168.169.10
```

```
FeatureType=1
```

19. VM を再起動すると、次のように、ライセンスサーバのライセンスクライアントの下にエントリが表示されます。



20. GATK および Cromwell ソフトウェアのダウンロードの詳細については、「Solutions Setup」セクションを参照してください。
21. GATK がオンプレミスで GPU を使用できるようになると、ワークフロー概要言語は「*」になります。wdl には、次に示すランタイム属性があります。

```

task ValidateBAM {
  input {
    # Command parameters
    File input_bam
    String output_basename
    String? validation_mode
    String gatk_path
    # Runtime parameters
    String docker
    Int machine_mem_gb = 4
    Int additional_disk_space_gb = 50
  }
  Int disk_size = ceil(size(input_bam, "GB")) + additional_disk_space_gb
  String output_name = "${output_basename}_${validation_mode}.txt"
  command {
    ${gatk_path} \
      ValidateSamFile \
      --INPUT ${input_bam} \
      --OUTPUT ${output_name} \
      --MODE ${default="SUMMARY" validation_mode}
  }
  runtime {
    gpuCount: 1
    gpuType: "nvidia-tesla-p6"
    docker: docker
    memory: machine_mem_gb + " GB"
    disks: "local-disk " + disk_size + " HDD"
  }
  output {
    File validation_report = "${output_name}"
  }
}

```

"次は終わりです"

まとめ

"前のバージョン： GPU セットアップ。"

世界中の多くの医療機関が、FlexPod を共通のプラットフォームとして標準化しています。FlexPod を使用すれば、医療機能を確実に導入できます。FlexPod と NetApp ONTAP には、業界をリードする一連のプロトコルを標準で実装できる機能が標準で搭載されています。特定の患者のゲノム研究の依頼の元にかかわらず、相互運用性、アクセシビリティ、可用性、およびスケーラビリティは、FlexPod プラットフォームに標準で

備わっています。FlexPod プラットフォーム上で標準化されると、イノベーションの文化は伝染しなくなります。

追加情報の参照先

このドキュメントに記載されている情報の詳細については、以下のドキュメントや Web サイトを参照してください。

- FlexPod Datacenter for AI / ML with Cisco UCS 480 ML for Deep Learning 』を参照してください

["https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_480ml_aiml_deployement.pdf"](https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_480ml_aiml_deployement.pdf)

- FlexPod Datacenter with VMware vSphere 7.0 and NetApp ONTAP 9.7 』を参照してください

["https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/fp_vmware_vsphere_7_0_ontap_9_7.html"](https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/fp_vmware_vsphere_7_0_ontap_9_7.html)

- ONTAP 9 ドキュメンテーション・センター

["http://docs.netapp.com"](http://docs.netapp.com)

- 即応性と効率性— FlexPod がデータセンターの最新化をどのように推進するか

["https://www.flexpod.com/idc-white-paper/"](https://www.flexpod.com/idc-white-paper/)

- 医療業界の AI

["https://www.netapp.com/us/media/na-369.pdf"](https://www.netapp.com/us/media/na-369.pdf)

- ヘルスケア向けの FlexPod で変革を促進

["https://flexpod.com/solutions/verticals/healthcare/"](https://flexpod.com/solutions/verticals/healthcare/)

- Cisco とネットアップが提供する FlexPod

["https://flexpod.com/"](https://flexpod.com/)

- ヘルスケア向けの AI と分析（ネットアップ）

["https://www.netapp.com/us/artificial-intelligence/healthcare-ai-analytics/index.aspx"](https://www.netapp.com/us/artificial-intelligence/healthcare-ai-analytics/index.aspx)

- 医療機関における AI スマートインフラの選択が成功を促進します

<https://www.netapp.com/pdf.html?item=/media/7410-wp-7314.pdf>

- FlexPod 9.8 を備えた ONTAP データセンター、Cisco Intersight 用の ONTAP ストレージコネクタ、および Cisco Intersight 管理モード。

<https://www.netapp.com/pdf.html?item=/media/25001-tr-4883.pdf>

- Red Hat Enterprise Linux OpenStack プラットフォームを搭載した FlexPod データセンター

["https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_openstack_osp6.html"](https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/flexpod_openstack_osp6.html)

バージョン履歴

バージョン	日付	ドキュメントのバージョン履歴
バージョン 1.0 以降	2021年11月	初版リリース

著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。