



KumoScale™ ソフトウェアでのクロスドメインデータレプリケーションについて データセンターにおける次世代NVMe-oF™ データ保護方式の提供方式

課題: データの安全性と可用性の維持

現代のデータセンターが抱える課題の一つに、いかに効率やパフォーマンスを犠牲にすることなく、データの安全性と可用性を維持することができるか、ということが挙げられます。

・ アプリケーションによって異なるデータ保護の要件

プライベートクラウドとサービスプロバイダーのデータセンターは、多様なことが特徴です。データセンターにおいてストレージ容量を最も消費しているアプリケーション(Webキャッシュ、分析パイプライン、機械学習など)の多くはデータ保護を必要としませんが、消失しないようにデータを守ったり絶えずデータが使えるようになっていることが重要なアプリケーションもあります。

・ 大規模環境ほど、障害ドメインも拡大する

データ保護に対する従来のアプローチは、SSD等ストレージ部品自体の故障に焦点を当てています。従来のアプローチではRAIDまたはレイジャーコーディングを使用しますが、ラック内のスイッチが数十台のクライアントマシンを処理し、電力供給サブシステムが数十台のラックに対応するような比較的小規模のシステムが対象で、拡張性に乏しい課題があります。データ保護に対する最新のアプローチでは、すべてがいつかは壊れることを前提としているので、大規模なインフラストラクチャの障害が及ぶ範囲を認識して、全体としてデータを守る冗長化構成が重要です。データセンターのハードウェアは多様ですので、障害ドメインの定義は単純で一般的なものでなければなりません。

・ パフォーマンスが重要

分散ストレージの主なパフォーマンス指標は、読み取りレイテンシです。帯域幅と1秒あたりの入出力 (IOPS) は並列処理により結局のところ、向上させることはできますが、レイテンシ(遅延)はそうはいきません。サーバー直結フラッシュストレージの低レイテンシと、ネットワークストレージの安全性と柔軟性のバランスをどう実現するか、がクラウドストレージアーキテクチャの中心的な課題なのです。

The NVMe™ over Fabrics (NVMe-oF™) なら、両方のメリットを生かしたソリューションを提供できます。しかし経験豊富なエンジニアなら、この可能性を無駄にしてしまうのはとても簡単なことを理解しています。トラフィックの負荷とレイテンシの関係は、どのシステムやサブシステムにおいても指数関数的に急激に高まるため、アプリケーションとストレージメディア間のボトルネックによって多大な遅延が生じることがあります。通常のIOオペレーションで、ストレージノード間に複数のネットワークホップが必要になるような設計は、避けることが重要です。

・ 人員不足のITチーム

効率的でパフォーマンスが高く安全なストレージの価値は疑いの余地がありませんが、クラウドストレージの実装には、これまでITリソースへの多額の投資が必要でした。一部のデータセンター事業者にとって、魅力あるROI (Return On Investment) のプライマリストレージとしての分離型フラッシュストレージは、依然として手が届きません。これは主に、過負荷のITチームが設計、展開、運用に必要なリソースを投入できず、特に障害発生時に適切に対応できないためです。ネットワークストレージシステムを選択する際に考慮すべき重要事項は、すでに使用されている管理および監視インフラストラクチャとの整合性と、運用スタッフに対しどの程度の時間が要求されるかです。

データ保護の選択肢:

現代のデータセンターにおいてデータを保護し高可用性を実現するには、データ保護のスキームと選択肢を理解することが必要です。

レプリケーション方式 vs RAID vs イレイジャーコーディング

データ保護のスキームは、故障から守るためまたは、ストレージメディアを分割配置するよう冗長性を確保することにあります。レプリケーション方式はデータ自体の複数のコピーを保存するだけですが、RAIDとイレイジャーコーディングは、符号化されたブロックを保存し、それらは障害発生時に損失したデータを再構築するために使用されます。これらの符号化システムは、2つの整数(k, n)によって形成されており、通常、n個のコードワードのうちk個以上が稼働していれば元のデータを回復するのに十分です。RAIDおよびイレイジャーコーディングシステムには、ストレージ効率を向上させると言われています。たとえば、(k, n) = (6, 8)の場合、コードの冗長性係数は33%であり、8つのストレージのうちの任意の2つの損失に耐えることができます。ただし、符号化されたシステムには、次のようにレプリケーション方式に比べて重要な弱点があります。

・ 小サイズデータの書き込みの問題

書き込み用の符号化された「ストライプ」を構築するために、(k, n)符号化されたデータ保護システムでは、k個のデータ用ブロックに、n-k個の符号化された同じサイズのブロックが追加されます。現代のストレージシステムでは、通常、サイズが4,000バイト(4kB)未満のI/Oの効率が著しく低下するため、チャンクサイズは通常4kBの倍数になります。(6, 8)の(k, n)符号および4kBブロックを使用すると、ストレージスタックは6ブロックのユーザーデータを8ブロック(32kB)の符号化ワードに組み立てる必要があります。その後8つのストレージデバイスに8つの4kB書き込みを実行します。

この形式のデータ保護は、オブジェクトストア、バックアップシステムおよび大規模オブジェクトのストリーミング用に最適化されたその他のストレージサービスに非常に適していますが、汎用ブロックストレージシステムでは致命的な欠陥が生じます。ホストが1つまたは数個の4kBブロックの同期書き込みを実行する場合、書き込み符号を生成するストレージスタックに十分なブロックを確保するには結果的に時間を要することになります。結果としてI/Oパフォーマンステストでは、非常に軽い負荷のシステムでも、異常に長い書き込みレイテンシが発生することがあります。実際のアプリケーションでは、システムはI/Oの完了を待って一時ハングアップし、完全な符号化ワードが入力されるまで書き込みキューに留まるため、システムが回復しない場合があります。この問題は、16kBのブロックですべてのデータ書き込みを実行するMySQL™などの「bツリー」タイプのデータベースで特によく見られます。

・ CPU使用率

符号化方式のもう1つの問題は、符号化ブロックを計算するためにストレージスタックがメモリからデータをCPUに読み出し、その後それらのブロックをメモリに書き戻す必要があることです。イレイジャーコーディングの計算は近年高度に最適化されていますが、この操作はデータバッファのコピーと同等です。NVMeプロトコルベースのI/OとLinux® I/Oスタックに対する最近の改良では、バッファコピーを削減または排除しようとしています。バッファコピーはCPUメモリの帯域幅とキャッシュリソースを使い果たし、I/Oパスにボトルネックを形成する原因となるからです。すべての書き込み操作を符号化すると、回避するのが難しいバッファコピーを繰り返し行うこととなります。これらの符号化方式のデータ保護スキームの欠点から、KumoScaleソフトウェアはその中核となるデータ保護メカニズムとしてデータレプリケーション方式を使用します。

クライアントベースとサーバーベースのレプリケーション

ここまでの説明では、データ保護エンコーディングは、ストレージの「バックエンド」ではなく、書き込み元で実行されることを前提としています。別の方法としては、生データをストレージノードの1つに転送し、データの一部だけを符号化し、そのストレージシステムとは別のノードや障害ゾーンに転送するやり方もあります(サーバーベースのレプリケーション)。このアプローチの主なメリットは、すべてのストレージ処理をストレージサブシステムに任せられ、コンピュータノードがデータ保護に関与しないようにするという点です。一方、クライアントベースのレプリケーションはパフォーマンスを劇的に向上させ、ストレージノードあたりのクライアント数を大幅に増やすことができるため、クライアントあたりのストレージコストを大幅に削減できます¹。

従来、ストレージシステムの費用対効果は、使用可能なストレージ容量の1バイトあたりの金額に基づいて測定されていました。しかし、NVMe標準の登場と費用対効果の高いフラッシュメモリの価格設定により、IOPSあたりのコストと読み取りレイテンシは同等に重要な指標となり、場合によってはより重要になっています。同時に、NVMe対応のSSDが提供する優れた容量とパフォーマンスにより、これらのSSDをコンピューティングノードから分離し、ネットワーク上で共有するという動きが急速に進んでいます。この環境では、高性能のネットワークストレージシステムに影響を与える要因を理解することが重要です。

・ ネットワーク上のボトルネック

KumoScaleストレージノード²では通常、2枚のデュアルポート100~200ギガビットイーサネット(GE)ネットワークインターフェイスカード(NIC)が搭載されています。この構成では通常PCIe® Gen3インターフェイスを使用しますが、スループットは各NICのPCIeインターフェイスによって、全二重で約22.7ギガバイト/秒(GB/s)に制限されます(変調方式とプロトコルオーバーヘッドのため)。実際の運用では、このようなストレージノードは数百または数千のVMにボリュームを提供することになるので、クライアントの総帯域幅は常にストレージノードの総帯域幅をはるかに上回ります。そのため、ストレージの効率を最大にすることは、ストレージノード上のネットワーク帯域幅の使用率を最大化することだと言えます。ストレージノードではなく、クライアントでのデータレプリケーションでは、ストレージのコストが大幅に低下します(GB/秒当たりのコストとIOPSあたりのコストに基づく)。レプリケーションがターゲット側で実行された場合、ストレージノード「A」に書き込まれたデータは、ストレージノード「B」と「C」に転送され、ノード「A」のネットワー

インターフェイスを3回通過することになります。これは、ネットワークインターフェイスでI/O容量を消費します（これは既にシステムパフォーマンスのボトルネックになっています）。クライアントでのデータレプリケーションは、この問題を解決します。

・動的マッピング

微少な書き込み時の問題を軽減するには多くの方法がありますが、最も一般的な方法は、クライアント側にある特定の論理ブロックアドレスに対応するのは、ストレージメディア上の固定部分であるという考え方を捨てることです。そうではなく、動的な再配置システムなら、次に来るデータを論理アドレスとは意識的に関連せず格納します。このような動的配置を行うには、ストレージシステム上の論理アドレスすべてをリストしたテーブルだけでなく、最後に書き込まれたデータの位置を持っている必要があります。

KumoScaleのクロスドメインデータレプリケーション

KIOXIA (旧東芝メモリ) のKumoScaleソフトウェアは、クロスドメインデータレプリケーション(CDDR: Cross-Domain Data Replication)と呼ばれる技術でユーザーデータを保護します。この機能はデータレプリケーションに基づいており、ストレージクラス単位(アプリケーション単位)でレプリカ数を柔軟に指定できます。CDDRでは、書き込み操作は同期的に並列処理され(各レプリカに転送され)、全てのレプリカへの書き込みが終わった時にアプリケーションに書き込み完了を通知します。



CDDR機能について

CDDRは、データセンターレベルのアプローチでドライブ障害やネットワークの分断に対する復元力を高めます。CDDRにより、KumoScaleソフトウェアユーザーは、ストレージノード、コンピュートノードおよびその他のリソースにタグを付けることで、任意の障害ドメインを特定できます。さらに、異なる障害ドメインにある複数のKumoScaleアプライアンスにボリュームレプリカを作成できます。このソリューションは、機器の障害や機器保守のためのダウンタイムというシナリオの両方に対応しており、ストレージクラスの仕様により、必要なアプリケーションにのみ適用できます。新しいクラウドネイティブアプリケーションは、データ復旧をアプリケーションレベルで行うため、ストレージレイヤーでのデータ復旧を必要としません。ストレージクラスのパラメータにより復旧方法の種類を設定することができるので、従来のアレイ型(データ保護の仕組みをローカルに実装)と比較してコストを大きく下げ、すべてのアプリケーションで同じ方法をとることができます。

CDDRソリューションは、ホストサーバーにあるLinuxの「md」モジュールを使用してレプリカ同期を行いますので、その成熟度と安定性、そして豊富な監視およびアラート機能を含む機能パッケージを活用できます。レプリケーションレイヤーは、ホストサーバー上のエージェントによって構成および監視されます。Kubernetesのコンテナ化されたフレームワークでは、このエージェントの実体はCSIドライバーです。ベアメタル環境では、このエージェントの実体はホスト上で実行されるジョブです。

レプリカに障害が発生すると、Linuxの「md」モジュールは更新されたデータのログを記録し始め、ボリュームのデータ保護レベルが低下したことを示すアラートを生成します。レプリカが回復すると、KumoScaleソフトウェアが自動的にレプリカを再接続し、「md」モジュールを介して最新の変更内容と同期します。レプリカが永久に切断されると、CDDRはトポロジに応じて別の場所にレプリカを再構築します。ボリュームが再構築されると、消失したレプリカの構成が削除されます。

ボリュームの割り当てと配置、レプリカの構成、およびネットワークファブリックを介したボリュームの接続は、すべてCDDRソリューションによって管理されます。通知とアラートは、事前に設定されたアプリケーションまたは電子メールアドレス(ベアメタルまたはVM環境の場合)およびsyslog(コンテナ化された環境の場合)に送信されます。

CDDRのメリット

KumoScaleソフトウェアによるデータレプリケーションへのアプローチには、選択可能な保護機能、トポロジー認識、パフォーマンス(帯域幅とレイテンシ)など、いくつかのメリットがあります。

・選択可能な保護機能

クラウドネイティブ環境のスケールアウト型アプリケーションには、多くの場合、独自の形式のデータ保護が含まれています。これには、MongoDB、Cassandra、AerospikeなどのKey-Valueタイプのデータベースが含まれます。これらのアプリケーションでは、ストレージレイヤーに冗長性を実装すると、復元力が大幅に向上することなくリソースだけが消費されます。一方、ほとんどの従来からあるアプリケーションは、ストレージが信頼できるという前提で設計されているため、ストレージレイヤーは復元力と可用性を提供する必要があります。KumoScaleソフトウェアは、復元力のレベルと配置ポリシーをストレージクラスごとに(アプリケーションごとに)個別に指定できるようにすることで、これらの要件を満たします。

・ **トポロジー認識**

大規模なデータセンターでは、物理インフラストラクチャは明確に定義された「障害ドメイン」に分割されます。電源供給、冷却、ネットワーク、その他のシステムの規模と設計によっては、障害ドメインはラック、ラックの列、さらにはデータセンター全体である可能性があります。小規模なレジリエントソリューションは、大きな障害ゾーンに対応出来ないため、誤った安心感をあたえます。KumoScaleソフトウェアは、ラベルを使用することで、KumoScale Provisioner Serviceに対し障害ゾーントポロジを簡単に特定できます。ラベルを参照するポリシーを作成すると、ボリュームレプリカが別の障害ドメインに配置されます。

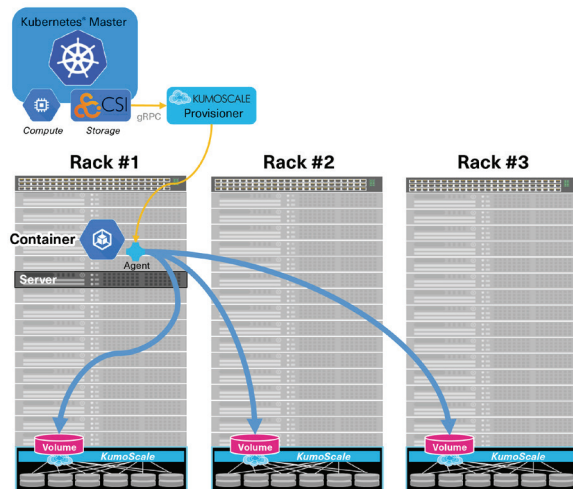
・ **パフォーマンス(帯域幅とレイテンシ)**

KumoScaleソフトウェアは、クライアントのLinux「md」レイヤーを介してレプリケーションを実行します。障害が発生したレプリカの全てのセットアップ、監視および修復は、コンテナ化された環境ではKumoScale Container Storage Interface (CSI)ドライバによって、ベアメタルまたは仮想マシン (VM) 環境の場合はクライアントエージェントによって処理されます。クライアントからレプリケーションを実行すると、データセンターのコンピューターノードの膨大なネットワーク帯域幅が活用され、ストレージノードで不要なボトルネックが発生するのを回避できます。その結果、ターゲット側のレプリケーションと比較して、1ドルあたりのIOPSと1ドルあたりの帯域幅が100%から200%向上します³。

CDDRの主要コンポーネント

KumoScale CDDRソリューションはいくつかのコンポーネントで構成されており、データ保護されたボリュームが割り当てられた時に発生する一連のイベントを図1に示します。これはKubernetesコンテナ環境の例です。:

KumoScale CDDRプロセス



Step 1:

Kubernetesオーケストレーターは、KumoScaleのCSIプラグインを介して3つのレプリカを持つボリュームを要求します。

Step 2:

CSIプラグインは、KumoScale Provisioning Serviceを呼び出して、冗長ボリュームを要求します。

Step 3:

KumoScale Provisioning Serviceはボリュームを定義し、NVMe-oFネットワークに直接アクセスするアプリケーション負荷のもとで動作するホストエージェントを構成します。

Step 4:

KumoScaleバックエンドはそれぞれ、ストレージクラス要件を満たす仮想ボリュームを提供します。

Step 5:

ホストエージェントは、すべてのレプリカのステータスを監視し、ネットワークまたはシステムに障害が発生した場合にボリュームのセルフヒーリング機能(自動回復機能)を開始します。

図1: KumoScaleソフトウェアのCDDRプロセス

・ **データセンターのプロビジョニング/オーケストレーションインフラストラクチャ**

データセンターのプロビジョニングまたはコンテナオーケストレーションは、アプリケーションホストのために復元力のあるボリュームを要求します。ボリュームにはストレージクラスにより必要なサービスクラスとレプリカ数が含まれます。

・ **CSIプラグイン**

Kubernetesコンテナ環境では、KumoScale CSIプラグインはKubernetesマスターの一部を形成します。ボリューム要求をKumoScale Provisioning Serviceに転送し、生成されたボリュームレプリカをネームスペースとして関連ホストに紐づけ、ホストからレプリカへのNVMe-oF接続を開始します。プラグインは、NVMeデバイス上にファイルシステムとマウントポイントも作成します。ベアメタル環境では、これらの機能はAnsible playbooksなどのプロビジョニング自動化ツールによって実行されます。

・ **KumoScale Provisioner Service**

KumoScale Provisioner Serviceは、ストレージクラスのトポロジパラメータとKumoScaleのSSD使用状況に基づいて、各ボリュームレプリカを特定のストレージノードおよびSSDにマッピングします。次に、関連するKumoScaleストレージノードにRESTコマンドを発行して、ストレージボリュームをプロビジョニングします。また、アプリケーションサーバ上のホストエージェントを設定します。

・ KumoScale ストレージノード

KumoScaleストレージノードは、KumoScale Provisioning Serviceからのコマンドに応答し、ボリュームを物理ドライブにマップします。また、I/Oコマンドを処理し、アクセス権を設定します。

・ クライアント(またはホスト)

クライアント(またはホスト)は、NVMe-oF標準を介して新しく作成されたネームスペースに接続します。クライアントのLinux「md」レイヤー経由のレプリケーションレイヤーは、ソリューションのデータベース部分を実行し、書き込みコマンドを複数のKumoScaleアプライアンスに複製します。また読み取り負荷を分散します。エージェント(ベアメタル環境またはVM環境)またはCSIドライバ(コンテナ環境)は、レプリケーションレイヤーの状態を監視し、syslogにイベントを生成します。

・ ロギング

データ保護の状態は、データセンターのsyslogに記録されます。通知とアラートはストレージ管理者に送信されます。

CDDRの主な機能

KumoScaleソフトウェアの主な機能には、イベントログとアラート、障害復旧、セルフヒーリング、計画的な一時停止、アプリケーションプログラミングインタフェース(API)によるインテグレーション統合などがあります:

・ イベントログとアラート

レジリエントな構成は継続的に監視されます。トリガーイベントが識別されると、メッセージがデータセンターログに送信されます。アラートは、データセンター管理者によってカスタマイズできます。たとえば、特定のトリガーイベントに対して電子メールが生成されたり、トリガーイベントの状態によりAPIが呼び出され、アラートシステムを起動します。

・ 障害復旧

何らかの理由でレプリカへの接続が失われると、レプリケーションレイヤーはボリュームへのすべての書き込み操作のログを開始します。消失したレプリカが再接続できた場合、再同期プロセスが開始され、残りのレプリカのデータを使用して、古いレプリカを最新の状態に戻します。このプロセスは、ネットワーク帯域幅とストレージパフォーマンスへの影響をコントロールするために抑制することができます。レプリカが完全に同期されると、サービスが再開されます。少なくとも1つのレプリカが機能している限り、アプリケーションは障害を認識しません。

・ セルフヒーリング

接続が失われたレプリカが設定されたタイムアウト時間内に復帰しない場合、そのボリュームは永続的に使用不可と宣言されます。これにより、使用可能なストレージノードに新しい空のボリュームがプロビジョニングされます。その新しいボリュームは、障害ゾントポロジの制約を含め、最初に作成されたストレージクラスのすべての仕様に従います。新しいボリュームはレプリカとして接続され、正常なレプリカのデータを使用して同期されます。障害が発生したレプリカが後で再表示された場合、そのレプリカに含まれるデータや論理的な設定情報はすべて消去されます。修復プロセスは、障害が検出されるとストレージ管理者によって手動で実行されるか、事前に設定したスクリプトにより自動的に実行されます。

・ 計画的な一時停止

データセンターの管理者は、アップグレードまたはメンテナンスのためにストレージノードをオフラインにすることがあります。このような場合は予期せぬレプリカの損失を避けるために、レプリカの構成を事前に変更することをお勧めします。CDDRソリューションは、復元可能な構成を維持しデータ保護レベルが劣化した状態を避けることによって、このシナリオをサポートします。

ストレージ管理者は、ダウンタイムに先立って新しいレプリカを追加するプロセスを開始します。その構成は「N+1」個のレプリカの状態になり、「N」個は安定状態の構成です。ラックの1つがダウンすると、構成は「N」個のレプリカに戻り、ボリュームは劣化状態になりません。

・ APIインテグレーション

KumoScaleソフトウェアの他の機能と同様に、CDDR APIは以下のように、データセンターで運用されている既存のインフラストラクチャとインテグレート出来るように設計されています:

- » CSIドライバーを介してコンテナ化された環境とKubernetesフレームワークをサポート
- » Ansible playbookと関連モジュールを使用してベアメタル環境をサポート
- » アラートやイベントログ等の顧客独自のロギングメカニズムをサポート
- » 統計用のテレメトリシステム (PrometheusやGraphiteなど) をサポート
- » RESTful APIを公開し、あらゆるプロビジョニングシステムに簡単に統合可能

まとめ

CDDRソリューションは、最新のデータセンターインフラストラクチャ向けに設計されています。トポロジーに配慮した環境で、複数のKumoScaleソフトウェアアプライアンスを活用し、各アプライアンス間でボリュームレプリカを配置します。これにより、ネットワーク、電源構成、SSDやKumoScaleソフトウェアなどの障害に対して復元力を発揮します。CDDRは、既存のデータセンターシステムや一般的なオーケストレーションフレームワークと連携するように構築されています。

KumoScaleソフトウェアの詳細については、<https://kumoscale.kioxia.com/jp> にアクセスしてください。

NOTES:

¹ クライアントからの書き込みごとにストレージへの2回の書き込みを必要とする2xレプリケーションに基づいています。2回目の書き込みが最初のターゲットから転送される場合、データは最初にそのターゲットに入り、その後再び送信される必要があります。これは、ストレージコストに悪影響を及ぼす可能性のある単純書き込みに対して2倍の帯域幅を消費することになります。

² 本技術概要は2020年8月時点で公表されたものです。

³ クライアントからの書き込みごとにストレージへの2回の書き込みを必要とする2xレプリケーションに基づいています。2回目の書き込みが最初のターゲットから転送される場合、データは最初にそのターゲットに入り、その後再び送信される必要があります。これは、単純書き込みに対して2倍の帯域幅を消費することになります。ターゲット側のレプリケーションと比較して、クライアント側のレプリケーションでは、1ドルあたりのIOPSと1ドルあたりの帯域幅が改善される可能性があります。

商標:

Ansibleは、米国およびその他の国におけるRed Hat, Inc.の登録商標です。Cassandraは、Apache Software Foundation (ASF)の米国およびその他の国における登録商標または商標です。Kubernetesは、米国およびその他の国におけるThe Linux Foundationの登録商標です。Linuxは、米国およびその他の国におけるLinus Torvaldsの商標または登録商標です。MongoDBは、MongoDB, Inc.の商標です。MySQLは、Oracleおよび/またはその関連会社の商標です。NVMeおよびNVMe-oFは、NVM Express, Inc.の米国またはその他の国における登録商標または商標です。PCIeは、PCI-SIGの登録商標です。Prometheusは、米国およびその他の国におけるThe Linux Foundationの商標または登録商標です。その他記載されている社名・商品名・サービス名などは、それぞれ各社が商標として使用している場合があります。

免責事項:

KIOXIA America, Inc.は、仕様および製品説明を随時変更することがあります。この技術概要に記載されている情報は情報提供のみを目的としており、技術的な誤り、漏れ、誤字脱字が含まれている場合があります。ソフトウェアまたはハードウェア構成の相違は、実際のパフォーマンスに影響を与える可能性があります。ここに含まれる情報は変更される可能性があり、多くの理由で不正確になる可能性があります。製品の変更、コンポーネントおよびハードウェアのリビジョン変更、新しいモデルおよび/または製品リリース、ソフトウェアの変更、ファームウェアの変更などを含みますが、これらに限定されません。KIOXIA America, Inc.は、本情報を更新またはその他の方法で修正または改訂する義務を負わないものとします。

KIOXIA America, Inc.は、本書の内容に関して一切の表明または保証を行わないものとし、この情報に表示される可能性のある誤り、漏れについて責任を負いません。

KIOXIA America, Inc.は、商品適格性または特定目的適合性に関する黙示の保証を明確に否認します。KIOXIA America, Inc.は、ここに含まれる情報の使用に起因する直接的、間接的、特別的、またはその他の結果的損害について、明示的にそのような損害の可能性を通知された場合でも、何人に対しても責任を負わないものとします。

© 2020 KIOXIA America, Inc. All rights reserved. 無断複写・転載を禁じます。このパフォーマンスブリーフに記載されている製品仕様、テスト内容、評価などの情報は、本書の発行日現在のものであり、ドキュメントが公開された日付時点で正確であると考えられていますが、予告なく変更される場合があります。ここに記載されている技術情報およびアプリケーション情報は、最新の該当するKIOXIA製品仕様に従います。