



Hewlett Packard
Enterprise

テクニカルホワイトペーパー

DATASTAX ENTERPRISE向けの HPE MOONSHOT

エッジからクラウドへのスケーリング – 抄訳



目次

エグゼクティブ サマリ	3
DataStax Enterpriseとは何か?	3
NoSQLデータベースの概念	4
HPE Moonshotを選択する理由	5
Moonshot上のDataStax Enterprise	6
HPE ProLiant m750 DSE パフォーマンス ベンチマーク.....	7
DSE Advanced Replicationによるエッジへのスケーリング	8
構成ガイドと構成明細	9
オペレーティングシステムと展開	9
ネットワーキング	9
ブレード上のストレージ	10
DataStax Enterprise	10
NoSQLBench	11
テスト済み構成のBoM	12
結論	12
付録-テストパラメータ	13
NoSQLBench参照のワークロード (cql-iot.yaml)	13
m750テストクラスターの参照構成 (cassandra.yaml)	15



エグゼクティブサマリ

機械学習や人工知能などのテクノロジーの最近の進歩により、ますます多くのデータが生成、保存、消費されています。データをデジタルエコノミーの基盤として採用する現代の企業は、増え続けるデータの取り込み、保存、処理、公開のために、高速で常時稼働可能なスケーラブルなプラットフォームを必要としています。

これらの目標を念頭に置いて設計されたApache Cassandra™の上に構築されたDataStax Enterpriseは、デジタルファーストの企業が今日の高速度常時稼働のデータ主導型経済で競争に勝つために必要なものを提供します。Cassandraは最先端の分散データベーステクノロジーになり、今日の最新のオンラインアプリケーションの多くを推進しています。継続的な可用性、高いスケーラビリティとパフォーマンス、強力なセキュリティ、操作のシンプルさを提供します。

DataStax EnterpriseとApache Cassandraが提供するユースケースの性質により、本番環境のデプロイメントは数千のノード以上に到達します。この規模の展開では、クラス最高のTCOを維持しながら、データセンターの内外のパフォーマンスとスケーラビリティを犠牲にしないように、基盤となるインフラストラクチャを慎重に計画する必要があります。

HPEはこのニーズに対応するために、顧客のアプリケーションスタックのコア、クラウド、エッジで業界最高のTCOを備えたスケールアウト高性能データクラスターの導入向けに設計および最適化されたMoonshotプラットフォームを提案します。

このドキュメントの目的は、HPE Moonshotで実行されるDataStax Enterprise (DSE) のリファレンスアーキテクチャについて説明し、パフォーマンスの数値、ベストプラクティスを提供し、MoonshotでオンプレミスおよびハイブリッドクラウドベースのDSEクラスターを実行するためのHPE-DataStaxソリューションの経済的メリットの概要を提供することです。

「お客様は、インストールが簡単なデータ処理ソリューションを介してビジネスに集中したいと考えています。Saagieと組み合わせたHPE Moonshotは完璧なソリューションです。」

– Jérôme Tredan、ゼネラルディレクター、Saagie

DATASTAX ENTERPRISEとは何か？

DataStax Enterpriseは、高速で常時稼働し、ゼロのダウンタイムに対処するように構築されており、人気のあるNoSQLデータベースApache Cassandraの商用バージョンを表しています。エンタープライズ向けのスケールアウトデータインフラストラクチャを備えたDataStax Enterprise (DSEとも呼ばれます)は、ゼロロックインとグローバルスケールを備えた、あらゆる場所のアクティブなゼロダウンタイムプラットフォームであらゆるワークロードを可能にします。

オープンソースのApache Cassandraと比較して、DataStax Enterpriseは、最大のインターネットアプリとFortune 100企業によって強化された運用の信頼性、監視、およびセキュリティレイヤーを追加します。

組み込みの高度なレプリケーションにより、真のエッジからクラウドへのスケーラビリティとデータベースモデルが可能になります。DataStax Enterprise 6.8は、クラウドネイティブおよびベアメタルのパフォーマンスを加速し、グラフから検索および分析まで、より多くのワークロードタイプをサポートし、KubernetesおよびAPIを使用してユーザーの生産性を向上させます。これにより、企業は、データの主権、可用性、スケール、俊敏性、およびすべてのユーザーによるアクセス性を必要とするアプリケーション、マイクロサービス、およびエクスペリエンスのための変革データアーキテクチャを構築できます。

DataStax Enterpriseについての詳細:

datastax.com/sites/default/files/content/datasheet/files/2020-04/5ypebYu4xqZOHIYsNU72fIDft6n1ziJdxsUxOdoO.pdf



NoSQLデータベースの概念

Microsoft SQLなどの従来のリレーショナルデータベースと比較して、NoSQLデータベースは、データの保存と管理に関して異なるアプローチを使用します。複数のテーブルをリレーションシップで相互に接続する代わりに、NoSQLデータベースのテーブルは相互に接続またはリンクされません。代わりに、テーブルは互いに独立しており、キースペースに編成されています。各キースペースは複数のテーブルで構成でき、レプリケーション係数や整合性レベルなどの特定の属性があります。Cassandra、したがってDSEは列ストアデータベースです。つまり、データは従来のSQLデータベースの行と比較して列に格納されます。つまり、同じテーブル内の行間でデータのフォーマットを変えることができます。

NoSQLの概念を活用することにより、企業組織は、トランザクション、ログ、またはIoTセンサーデータなどの大量のデータを格納する際の速度、機敏性、およびパフォーマンス効率の向上から恩恵を受けます。また、リングトポロジを使用してデータが複数のノード(クラスターとも呼ばれます)に分散されるため、柔軟性と線形スケーリングが大幅に向上します。他のリレーショナルデータベースシステムとは異なり、このトポロジーの拡張はシンプルで無停止であり、新しく追加されたノードの完全な追加パフォーマンスを利用できます。

データを複数のノードに分散するために、NoSQLデータベースシステムはデータをパーティションに編成します。各パーティションには、テーブルの定義されたパーティション列の内容に基づいてハッシュ生成された番号であるパーティションキーがあります。クラスターのトークン範囲は $2^{127}-1$ です。この範囲は、クラスター内の使用可能なすべてのノードに分割されます。つまり、各ノードは特定の範囲をカバーします。データセットのハッシュ値(パーティションキー)が含まれるノードは、そのデータを格納お処理します。たとえば、新しいデータセットのハッシュパーティションキーが45の場合、図1に従って、ノード番号1がそのデータを格納します。

上記のように、キースペース作成の一部として、複製係数(RF)を指定する必要があります。これは、データを保存する必要があるノードの数について説明します。

図1では、RFは3に設定されています。つまり、すべての書き込みは、プライマリ以外の2つの追加ノードに対して行う必要があります。NoSQLクラスターから読み取る場合、データをクライアントに渡す前に読み取る必要のあるノードの数から、整合性レベルが指定されます。これにより、一貫性のあるデータがクライアントに確実に返されます。整合性レベルは1から多数、またはクォーラムに設定できます。これは、データを保持するノードの大部分が同じデータを返す必要があることを意味します(たとえば、3つのノードのうち2つ)。

NoSQLクラスターと対話するには、特定のクエリ言語(Cassandraクエリ言語など)とクラスター対応ドライバーが必要です。リクエストされたデータを保存または提供していなくても、NoSQLクラスター内の任意のノードを使用してクラスターに接続できます。これにより、その特定のノードがリクエストコーディネーターになり、RFレベルと整合性レベルが確実に満たされるようになります。

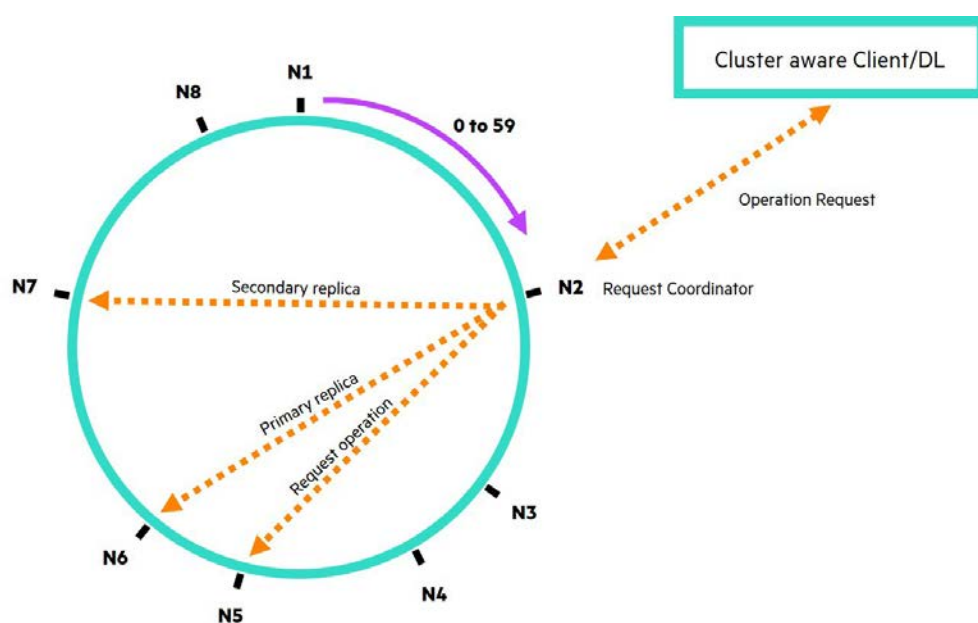


図1. NoSQL Clusterトポロジーの概要



HPE MOONSHOTを選択する理由

HPE Moonshot Systemは、1つのシンプルな設計理念に基づく革新的なアーキテクチャを使用します。専用サーバーを適切なワークロードに合わせて、環境に最適な結果を提供します。従来のサーバーは、単一のシャーシ内の管理、ネットワーク、ストレージ、電源コード、冷却ファンなどの専用コンポーネントに依存しています。一方、Moonshotシステムはこれらのシャーシコンポーネントを共有し、コンパクトな4.3Uシャーシで最大45のワークロード最適化ProLiantサーバーブレード(それぞれ最大16のIntel®Xeon®コア、128 GB RAMおよび16 TB NVMe SSDストレージを提供)をサポートできます。これにより、運用コストを削減しながら、より小さなフットプリントからより大きな収益を生み出すことができます。これは、DataStax EnterpriseやApache Cassandraなど、垂直方向にスケーラブルなワークロードを見るときに特に当てはまります。

図2は、プラグインブレードのフォームファクターで最大45の個別サーバーをホストする4.3U Moonshotシャーシを示しています。複数の種類のブレードを備えたプラットフォームコンピューティング機能は、特定のワークロードの経済に完全に一致するように簡単に調整でき、コンピューティング、電力消費、およびコストの理想的なバランスを実現します。

HPE Moonshotは、DataStax Enterpriseクラスターを大規模に実行するためのクラス最高のプラットフォームです。



図2。DataStax Enterprise向けのMoonshotブレードファミリーを備えたHPE Moonshot 1500シャーシ

- 理想的なコンピューティングフットプリント:** 4つから最大16個の物理Intel Xeonコアを搭載したブレードにより、HPE Moonshotは、あらゆるパフォーマンスレベルのベアメタルDSEおよびCassandraクラスターを大規模に展開するのに最適な選択肢になります。「one-blade-per-one DSEノード」展開モードは、ノードごとのパフォーマンスのスイートスポットになりますが、仮想化またはコンテナ化された環境でDSE/Cassandraを実行する場合の運用上の複雑さ、ハイパーバイザーに関連する容量ペナルティ、または従来の大規模サーバープラットフォームでDSE/Cassandraベアメタルを実行する場合に典型的な「未活用の問題」はありません。
- 究極の密度と拡張性:** 4.3Uシャーシに最大45のベアメタルノードがあり、MoonshotでDSEおよびCassandraクラスターを実行することは、業界で最も密度の高い展開です。ノードごとのデュアル10GbEネットワークとシャーシに組み込まれたノード間スイッチングにより、HPE Moonshotは、DSE/Cassandraクラスターのノード間通信にクラス最高のレイテンシを提供すると同時に、追加の電源ケーブルやネットワークケーブルを使用せずにブレードをより多くプラグインすることで、クラスターのスケールアウトを容易にします。
- エッジへの拡張:** EL4000シャーシでは、最大4つのHPE Moonshotブレードを使用し、1つのアーキテクチャを維持しながら、ガソリンスタンド、倉庫、産業プラントなど、データセンターの外部の任意の場所でブレードを操作できます。
- 最適化されたブレードアーキテクチャ:** 各Moonshotブレードには、ストレージ用のNVMeモジュールが装備されています。これにより、DSE/Cassandraノードの比類のないI/Oが可能になり、クラスターの全体的な遅延が大幅に減少します。デュアルMellanox 10 Gbps低レイテンシネットワークとDSE/Cassandraのベアメタル展開を組み合わせると、HPE MoonshotはDSE/Cassandra展開にクラス最高のパフォーマンスを提供します。
- 最適化されたTCO:** HPE Moonshotプラットフォームは、DSE/Cassandraクラスターの導入と運用のための最低のCAPEXとOPEXを提供します。電力とコストが最適化されたIntel Xeon CPUラインに基づくHPE Moonshotブレードラインアップは、同様のパフォーマンスフットプリントを持つ従来のサーバーシステムと比較して、ノードあたりの消費電力が2分の1になります。

注記: Moonshot ProLiant m750ブレードと他のMoonshotブレード(m710x-Lやm510など)を1500シャーシ内で混在させることは、Chassis Manager 2(CM2)の制限により、現在サポートされていません。

MOONSHOTのDATASTAX ENTERPRISE

HPE Moonshotは、3000を超えるDSEおよびCassandra運用ノードを実行し、現在までに世界中で10 PBを超えるデータを保存しているため、スケーラブルなDSEおよびCassandraの導入においてクラス最高のパフォーマンスとTCOを実証したプラットフォームです。DataStaxとHPEの密接な関係により、HPE Moonshotシステムでの最適な互換性とパフォーマンスが保証されます。

図3は、ブレードのさまざまなフレーバーに基づいて、HPE Moonshotによって提供されるさまざまな使用例を示しています。

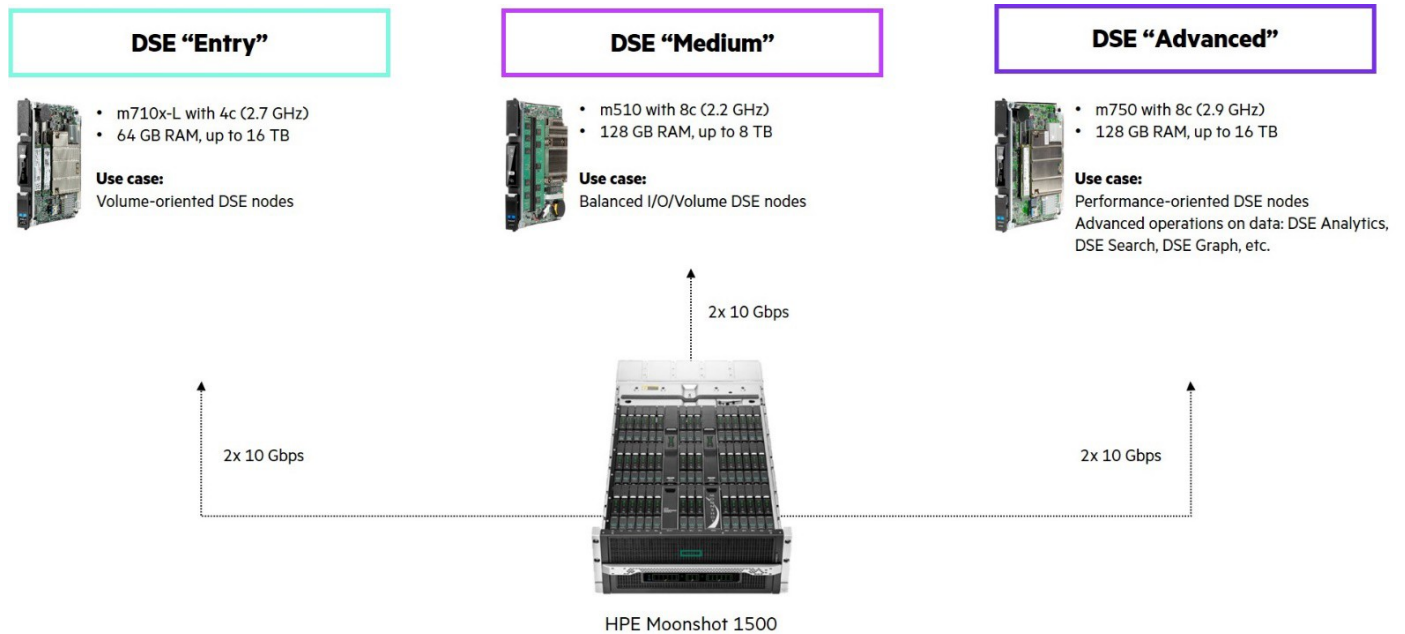


図3。HPE MoonshotのDSEパフォーマンス階層

- DSEの「エントリー」は、DSEクラスターが大量の低速データを格納しているユースケースを示しています。これは、ノードあたり最大16 TBのNVMeストレージを備えたm710x-Lブレードによって提供され、ブレードあたり最大4つの物理CPUコアを提供すると同時に、DSEインフラストラクチャに低コストのソリューションを提供します。
- DSE「ミディアム」は、I/Oからボリュームへのバランスのとれたユースケースを表し、各DSEノードはブレードあたり最大8つの物理コアによってサービスされます。
- DSE「アドバンスド」は、基礎となるデータベース（DSE Analytics、DSE Searchなど）の上でDSEの追加の付加価値機能を実行する可能性がある、高性能のDSEノードクラスターを示しています。

概説されているMoonshotパフォーマンス階層は、幅広いワークロードで最高のパフォーマンスとTCO比を保証します。DataStaxとMoonshotの両方のアーキテクチャでは、環境の拡大に応じて独立したブレードを追加することにより、きめ細かなスケーリングが可能になり、リレーショナルデータベースとは対照的に、DataStax Enterpriseは新しく追加されたクラスターノードを最大限に活用できます。

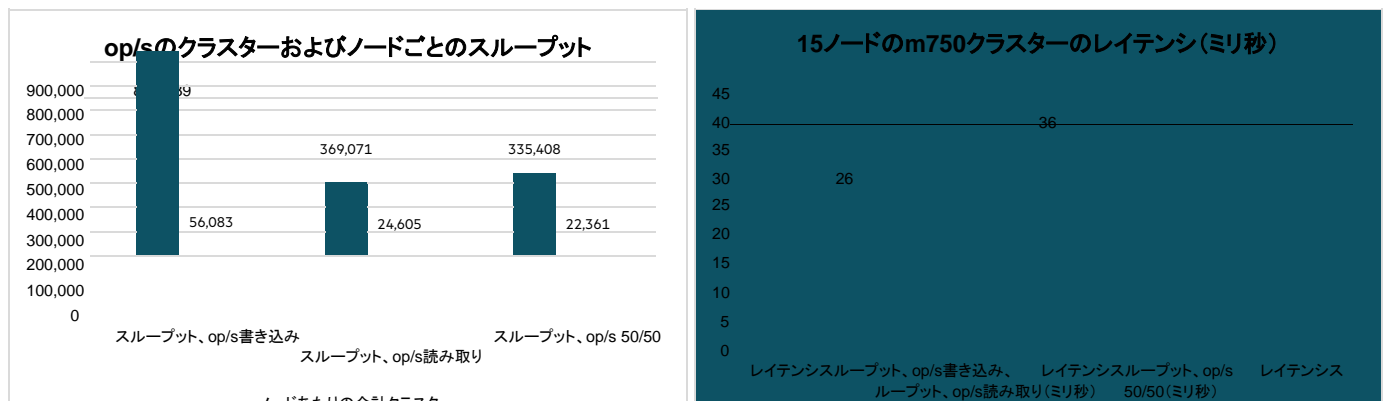


HPE ProLiant m750 DSEパフォーマンスベンチマーク

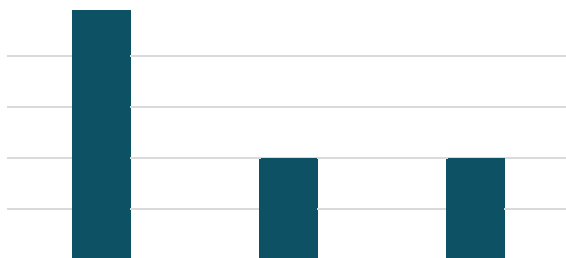
m750は、DSEおよびNoSQLクラスターに新しいレベルのパフォーマンスを提供します。ブレード上の物理DSEインスタンス、高周波コア、低レイテンシNVMeドライブは、低レイテンシの組み込みネットワークファブリックとペアになっており、分散データベースを新しいパフォーマンスレベルに加速します。

m750は、DSEおよびNoSQLクラスターに新しいレベルのパフォーマンスを提供します。

HPEは、単一の1500シャーシにある15ノードのm750クラスターでテストを実施し、NoSQLBenchを使用して一般的なIoTワークロードをシミュレーションしました。



HPE Moonshotは、低レイテンシを維持しながらノードあたり56K IOPSに達するスループットで、最適なTCOを備えた適切なサイズのベアメタルDSE展開の利点を実証します。独立したブレードアーキテクチャにより、パフォーマンスは単一または複数のシャーシにわたって直線的に拡張されます。



これは、単一のHPE Moonshotシャーシの5U展開のフットプリントに基づく印象的な数値に変換されます:

- 最大240万の書き込みワークロードIOPS
- 最大100万の混合ワークロードIOPS
- 最大90万の読み取りワークロードIOPS

ベンチマークデータがドライブに書き込まれ、現実的な実際の動作がシミュレーションされることを確認するために、クラスターが立ち上がった後にテストが行われました。

テスト済みの構成の詳細については、このドキュメントの「構成と付録」セクションで説明しています。



DSE Advanced Replicationによるエッジへのスケーリング

HPEは、Moonshot上のDSEのスケーラブルな集中型オンプレミスおよびハイブリッド展開を補完し、DSE Advanced ReplicationおよびHPE Edgelineプラットフォームに基づくEdge-to-Coreアーキテクチャにより、DSE展開に柔軟性を追加します。

製造、石油、ガス、小売、その他の業界の多くの企業が、散在的な接続を使用して小規模な遠隔地に大量に分散されたデータをサポートしながら、サイト間のデータの可用性と一貫性を維持できるデータベーステクノロジーを求めています。DSEはDSE Advanced Replicationメカニズムでそのニーズに対応し、HPEはリモートエッジロケーションの課題に対処するために特別に設計された、エッジ最適化された独自のエッジラインプラットフォームを提供します。

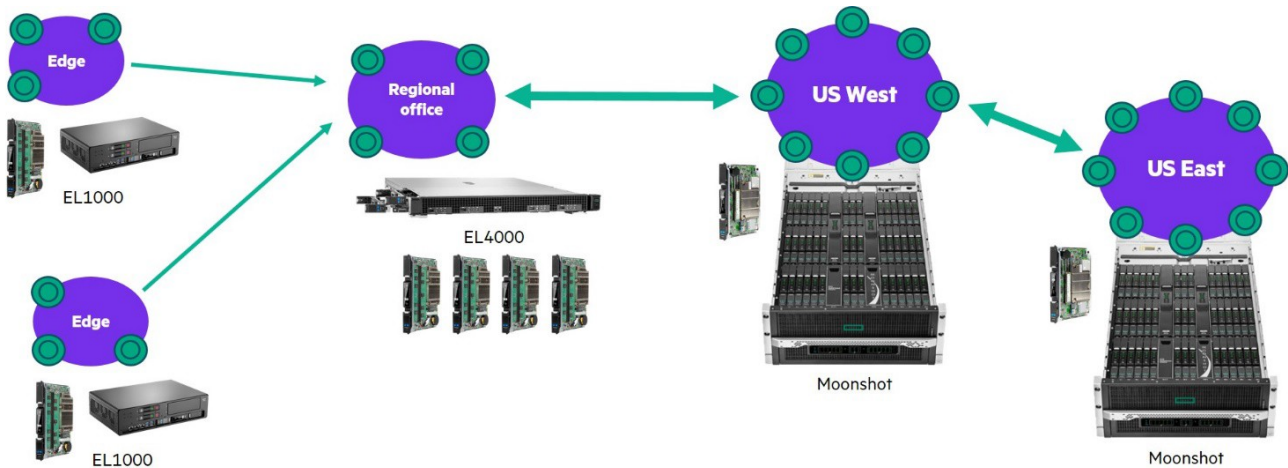


図4. DSE Advanced Replication、HPE Moonshot、およびEdgelineを使用した真のエッジツーコアDSE展開

HPEは、Edgeline EL1000およびEL4000プラットフォームでそのアーキテクチャを実現します。EL1000およびEL4000プラットフォームは、完全なリモート管理機能を備えた完全に堅牢な安全なエンクロージャとして設計されていますが、HPE Moonshotプラットフォームと同じコンピューティングブレードを利用しています。これにより、適切なサイズのベアメタルDSEノードと同じ比類のないパフォーマンスがコアからエンタープライズのエッジまでもたらされ、エッジとコア間で動作、スペアパーツ管理、インフラストラクチャ計画がシンプルかつ一貫性を保ちます。

エンタープライズインフラストラクチャのコアで実行される1500シャーシにより、お客様は、単一のプラットフォームとアーキテクチャを複数の場所とエッジに垂直に拡張することでメリットを得ることができ、DataStax Enterpriseのパワーをデータの生成、取り込み、処理にもたらしめます。

ハイティアm750 Advanced DSEノードを使用すると、DSE Analytics、DSE Search、DSE GraphなどのDataStaxの追加機能がサポートされます。

- **DSE分析:** Apache Spark™に基づいて、お客様はアドホックレポートを簡単に生成し、パーソナライズされたクライアントをターゲットにして、データのリアルタイムストリームを処理できます。分析ツールセットを使用すると、オペレーターはコードを1回記述すれば、リアルタイムとバッチの両方のワークロードに使用できます。
- **DSE検索:** お客様がデータをすばやく見つけて最新の検索エクスペリエンスをユーザーに提供できるようにし、製品カタログ、ドキュメントリポジトリ、アドホックレポートエンジンなどの機能の作成を支援します。
- **DSEグラフ:** 高速のデータストレージとトラバーサル、ゼロダウンタイム、および複雑で異種の関連データセットのリアルタイム分析のために最適化された分散グラフデータベースです。大規模なデータセットにスケーリングし、トランザクションワークロードと分析ワークロードの両方を実行できます。

HPE MoonshotおよびEdgelineのスケーラビリティとパフォーマンスと組み合わせた完全なDSE機能セットにより、顧客はいつでもどこでもデータの洞察を得ることができます。



構成ガイドと構成明細

オペレーティングシステムと展開

HPE Moonshotは、単一のデータセンターシャーシ内に45の専用計算ノードを提供します。これにより、オペレーティングシステムの展開と大規模な管理に関する問題が発生する可能性があります。HPEは、Moonshotブレードを大規模に管理および展開するためのPerformance Cluster Managerを提供しています。これには、Kickstart、AutoYaST、Preseedなどの自動インストール機能が含まれます。さらに、Performance Cluster Managerには統合されたゴールデンイメージキャプチャおよび展開機能があり、顧客は参照イメージを作成し、それをキャプチャして、同時に任意の数の同一のMoonshotブレードに配布できます。

m750 RHEL 7.6とHPE Performance Cluster Manager v1.0でのHPEの内部テストでは、ゴールデンイメージ配布アプローチが使用されました。

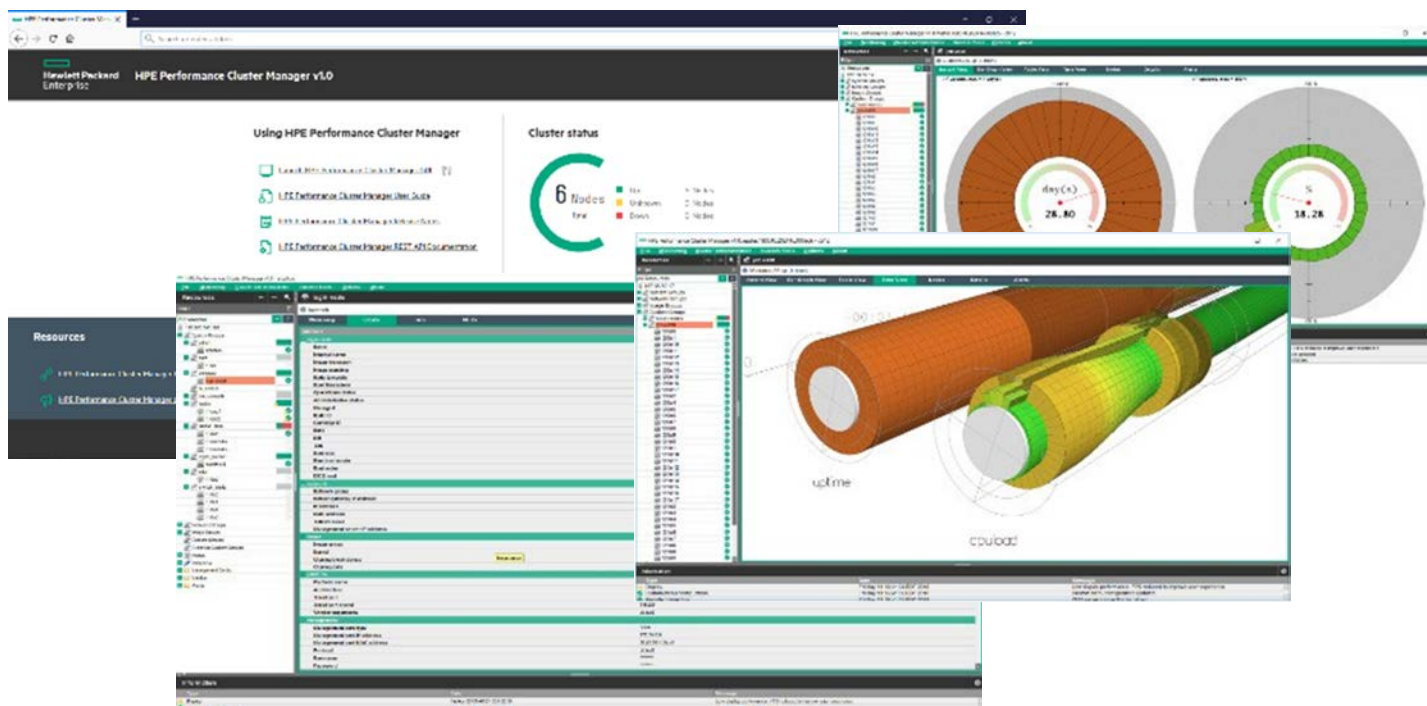


図5。Moonshot m750ブレードを管理するHPE Performance Cluster Manager

もちろん、オペレーティングシステム(OS)をHPE Moonshotに展開する方法は非常に多くあります。組織のベストプラクティスに応じて、他のイメージ展開または構成管理ユーティリティを使用できます。

ネットワーキング

ソリューションを構成する場合、HPE Moonshotシャーシアップリンクモジュールは、40GbE QSFP+ポートを使用してトップオブブラック(TOR)スイッチに直接接続する必要があります。TORスイッチに40GbEポートがない場合は、4QSFP+ アップリンクモジュールの代わりに16SFP+ アップリンクモジュールを使用し、アップリンクモジュールからTORスイッチへの複数の10GbEリンクが実行されているリンクアグリゲーショングループ(LAG)用にスイッチを構成できます。マルチシャーシ構成では、各アップリンクモジュールを上記と同様の方法でTORスイッチに接続する必要があります。または、シャーシアップリンクモジュールと一緒にスタックすることもできます。詳細については、Moonshotネットワーキングガイドを参照してください。



ブレード上のストレージ

各HPE Moonshot ProLiant m750サーバーブレードは、最大4つのNVMeドライブと1つのSATAドライブを実行できます。テスト構成では、HPEは1つの240 GB SATAドライブを使用してOSをホストし、2つの個別のNVMeドライブをデータおよびキャッシュストレージ用に使用しました。

以下の構成が使用されました:

ドライブ	容量	種類	バス幅	マウントポイント	使用用途
SATA 1	240 GB	混在使用	x1	/sda	OSドライブ
NVMe 1	960 GB	混在使用	x4	/nvme1	データドライブ
NVMe 2	960 GB	混在使用	x4	/nvme2	キャッシュドライブ

DataStax Enterprise

DataStax Enterpriseは、次のいくつかの方法で配置できます: tarball経由、OS固有のパッケージ(rpm、deb)経由、またはまたはDataStaxライフサイクルマネージャー(LCM)経由。これには、視覚的な管理コンソールであるDataStax OpsCenterも含まれ、複数のシャーシ、ラック、および場所にまたがる複雑なクラスターと展開の制御と監視を簡素化します。LCMとOpsCenterは、管理者、アーキテクト、および開発者が一元化されたダッシュボードからシステムを一目で把握できるようにすることで、Apache CassandraおよびDataStax Enterpriseクラスターの管理を容易にします。

OpsCenterとLCMはシームレスにインストールされ、システムオペレーターは、どんな複雑なワークロードでも、Webブラウザーから簡単に監視および管理できる柔軟性を備えています。

LCMとOpsCenterに加えて、DataStaxは、高度なモニタリングとレポートのための優れたGrafanaダッシュボードとPrometheus統合を提供します。これらのツールを使用してDataStax環境を管理することを強く推奨しています。

HPEは、LCMおよびOpsCenterとともにDataStax Enterprise 6.8.0を使用しました。インストールビットと追加のコンポーネント(Cassandraクエリ言語ドライバー、バルクデータローダー、クエリツールなど)は、[.datastax.com/download](http://datastax.com/download)をご覧ください。



図6. m750テストクラスターを示すDataStax OpsCenterおよびGrafanaダッシュボード



DataStax LCMとOpsCenterをインストールしてDataStax Enterpriseを展開するには:

1. DataStax Enterpriseのドキュメント、tarballファイルとドライバー、OpsCenterとLCMをダウンロードするには、datastax.com/download にアクセスしてください。
2. OSパッケージマネージャーが設定されていること、およびオペレーティングシステムまたはゴールデンイメージでプロキシおよびインターネットゲートウェイに必要なパラメータを設定していることを確認してください。
3. OSに適用可能なDataStaxリポジトリにアクセスするようにパッケージマネージャーを構成し、単一のマシンにOpsCenterをインストールして起動します。
 - a. **Red Hat® Enterprise Linux®:** docs.datastax.com/en/install/6.8/install/opsclInstallRHEL.html
 - b. **Debian:** docs.datastax.com/en/install/6.8/install/opsclInstallDeb.html
 - c. **その他のLinux:** docs.datastax.com/en/install/6.8/install/opsclInstallTar.html
 - d. **Docker:** docs.datastax.com/en/install/6.8/install/opsclInstallDocker.html
4. インストールして開始したら、psCenter webGUI:opscenter-host:8888/に移動します
5. 「新しいクラスターの作成」を選択し、プロンプトに従って、システムのクラスター名、データセンター、IPアドレスの詳細、およびシステムの資格情報を入力します。リポジトリ(オンラインまたはオフライン)と構成プロファイルも作成する必要があります。HPEが使用する構成プロファイルについては、付録、cassandra.yamlを参照してください。より詳細な手順については、docs.datastax.com/en/opscenter/6.8/opscl/LCM/opsclInstallDSE.htmlをご覧ください。
6. データディレクトリの場所など、環境に適用可能な特定の設定がないか、アプリケーションで必要な場合はネイティブトランSPORTを有効にするために、構成プロファイルを確認してください。
7. 展開の準備ができたなら、クラスターに移動し、クラスターを選択して、その横にある3つのドットをクリックします。「インストール」をクリックしてプロンプトを確認します。確認すると、アクティブなジョブへのリンクが表示されます。
8. リンクをクリックするか、ジョブに移動します。タスクが実行されているのがわかります。ジョブの矢印をクリックすると、特定のノードのサブタスクにドリルダウンでき、現在のジョブのステータスとアクションが表示されます。これは、エラーが発生した場合に特に役立ちます。
9. パッケージはインターネット経由でインストールされ、クラスターは数分以内に機能します。
10. インストールすると、クラスターは自動的にOpsCenter Monitoringに表示され、LCMから直接アクセスできます。

注記

このセクションで提供されるすべてのリンクは、このドキュメントの作成日現在で最新のものです。外部リンクはいつでも予告なく変更されることがあります。HPEは、リンクが最新であること、セキュリティ修正を組み込んだこと、または最新バージョンを指すことを保証しません。

新しいMoonshot ProLiant m750ブレードをベンチマークするために、HPEはこちらで入手できるNoSQLBench (DSBenchのオープンソースの進化)を使用しました: datastax.com/blog/2020/03/nosqlbench

NoSQLBenchは、NoSQL環境向けの他のベンチマークツールと比較して、実際のシステムパフォーマンスの評価に役立つ、より生産的な現実的なワークロードシナリオを提供します。NoSQLBenchのもう1つの利点は、大規模システムのベンチマークを行う際に提供されるスケーラビリティと柔軟性です。HPEは、最大30ノードを使用して、m750クラスターに対してNoSQLBenchを実行し、システムを飽和させ、実際のストレスシナリオをシミュレーションしました。

エッジからクラウドまで画期的なパフォーマンスを提供するMoonshotの機能を強調するために、HPEはNoSQLBenchを使用してIoTサンプルアクティビティを実行しました。このベースラインの内容は、このドキュメントの付録に記載されています。

読み取り、混合、書き込みのベンチマークの前に、システムは2 TBのデータで増加しました。キースペースは実行と実行の間にページされませんでした。



テスト済み構成のBoM

この構成明細は、45アドバンスドノードのDSEベアメタルクラスターのHPE Moonshot構成の例を表しており、250万以上のIOPS(書き込み集中型)に達し、DSEデータに最大45 TBのrawストレージを提供します。

数量	部品番号	説明
1	P18680-B21	HPE Moonshot 1500 Configure-to-order 2.0 Chassis
2	704652-B21	HPE Moonshot-4QSFP+ Uplink Module Kit
2	704654-B21	HPE Moonshot-45XGc Switch Module Kit
45	P17342-B21	HPE ProLiant m750 E-2286M 5.0GHz 8-core 45W Configure-to-order Blade Server
180	P22156-B21	HPE 32GB 2Rx8 DDR4-2933Y-T Kit
45	866844-B21	HPE 240GB SATA M.2 2242 Solid State Drive Field Upgradable Kit
90	P05892-B21	HPE Edgeline 960GB NVMe x4 Lanes Mixed Use M.2 22110 3yr Wty Extended
4	684532-B21	HPE 1500W Common Slot Platinum Plus Power Supply Kit
1	681254-B21	HPE 4.3U Server Rail Kit
1	681260-B21	HPE 0.66U Spacer Blank Kit
45	E6U64ABE	HPE iLO Advanced Electronic License with 3yr Support on iLO Licensed Features

注記

部品番号は変更される場合があります。構成明細にはサポートオプションやその他のラックおよび電源要件は含まれていません。注文に関して質問がある場合は、詳細についてHPE営業担当者または認定リセラーにお問い合わせください。
hpe.com/us/en/contact-hpe.html

結論

DataStax EnterpriseとHPE Moonshotの組み合わせは、両方のアーキテクチャがパフォーマンスや効率を犠牲にすることなく直線的かつきめ細かくスケールするので、非常にうまく連携しています。データベースは、そのアーキテクチャに関係なく、非共有環境でベアメタルを実行するときに常に最高のパフォーマンスを発揮します。これまで、これは非効率的で管理が複雑であることが証明されており、OPEXコストが高くなります。Moonshotは、その小さなワークロード固有のコンピューティングブレードにより、並外れたパフォーマンスときめ細かいスケールを可能にすると同時に、電力、冷却、ネットワーク、および管理の複雑さを軽減することでOPEXコストを削減します。

ベアメタル展開は、HPE Performance Cluster Managerを使用してブレードを迅速に展開し、DataStax Lifecycle Managerを使用してノードを構成、保守、およびアップグレードするよりも簡単で、しかも超高速のNVMeストレージと物理CPUコアおよび速度の恩恵を受けています。

小規模なDevOps環境から大規模なマルチシャーシ展開まで、Moonshotがカバーし、ブレードのさまざまな層を使用して適切な量のコンピューティングおよびストレージリソースを提供します。アプリケーションとユースケースがエッジへのスケールアウトを要求する場合、HPE MoonshotブレードをベースとするHPE Edgelineファミリは、必要な場所でいつでも同じパフォーマンスとアーキテクチャを提供します。

全体として、Moonshotは次世代NoSQL展開を強化するための理想的なプラットフォームです。

詳細はこちら

<https://www.hpe.com/jp/ja/servers/moonshot.html>

付録—テストパラメータ



NoSQLBenchの参照ワークロード(cql-iot.yaml)

```
# dsbench -v run type=cql yaml=baselines/cql-iot tags=phase:schema host=dsehost bindings:
  machine_id: Mod(<<sources:10000>>); ToHashedUUID() -> java.util.UUID sensor_name:
  HashedLineToString('data/variable_words.txt')
  time: Mul(<<timespeed:100>>L); Div(<<sources:10000>>L); ToDate() cell_timestamp:
  Mul(<<timespeed:100>>L); Div(<<sources:10000>>L); Mul(1000L) sensor_value:
  Normal(0.0,5.0); Add(100.0) -> double
  station_id: Div(<<sources:10000>>);Mod(<<stations:100>>); ToHashedUUID() -> java.util.UUID data:
  HashedFileExtractToString('data/lorem_ipsum_full.txt',800,1200)
blocks:
- tags:
  phase: schema
  params:
  prepared: false
  statements:
  - create-keyspace: |
    create keyspace if not exists <<keyspace:baselines>>
    WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '<<rf:1>>'} AND
    durable_writes = true;
    tags:
    name: create-keyspace
  - create-table : |
    create table if not exists <<keyspace:baselines>>.<<table:iot>> ( machine_id
    UUID, // source machine
    sensor_name text, // sensor name
    time timestamp, // timestamp of collection
    sensor_value double, //
    station_id UUID, // source location
    data text,
    PRIMARY KEY ((machine_id, sensor_name), time)
    ) WITH CLUSTERING ORDER BY (time DESC)
    AND compression = { 'sstable_compression' : '<<compression:LZ4Compressor>>' } AND
    nodesync={'enabled': 'true'}
    AND compaction = {
    'class': 'TimeWindowCompactionStrategy',
    'compaction_window_size': <<expiry_minutes:60>>,
    'compaction_window_unit': 'MINUTES', 'split_during_flush': true
    };
    tags:
    name: create-table
  - truncate-table: |
    truncate table <<keyspace:baselines>>.<<table:iot>>; tags:
    name: truncate-table
- tags:
  phase: rampup
  params:
  cl: <<write_cl:LOCAL_QUORUM>>
  statements:
  - insert-rampup: |
    insert into <<keyspace:baselines>>.<<table:iot>>
```



```

(machine_id, sensor_name, time, sensor_value, station_id, data)
values ({machine_id}, {sensor_name}, {time}, {sensor_value}, {station_id}, {data}) using timestamp
{cell_timestamp}
idempotent: true tags:
  name: insert-rampup
- tags:
  phase: verify type:
    read
  params: ratio:
    1
    cl: <<read_cl:LOCAL_QUORUM>>
  statements:
    - select-verify: |
      select * from <<keyspace:baselines>>.<<table:iot>>
      where machine_id={machine_id} and sensor_name={sensor_name} and time={time};
      verify-fields: "*", -cell_timestamp"
      tags:
        name: select-verify
- tags:
  phase: main type:
    read
  params:
    ratio: <<read_ratio:1>>
    cl: <<read_cl:LOCAL_QUORUM>>
  statements:
    - select-read: |
      select * from <<keyspace:baselines>>.<<table:iot>>
      where machine_id={machine_id} and sensor_name={sensor_name} limit
      <<limit:10>>
      tags:
        name: select-read
- tags:
  phase: main type:
    write
  params:
    ratio: <<write_ratio:9>>
    cl: <<write_cl:LOCAL_QUORUM>>
  statements:
    - insert-main: |
      insert into <<keyspace:baselines>>.<<table:iot>>
      (machine_id, sensor_name, time, sensor_value, station_id, data)
      values ({machine_id}, {sensor_name}, {time}, {sensor_value}, {station_id}, {data}) using timestamp
      {cell_timestamp}
      idempotent: true
      tags:
        name: insert-main

```



m750テストクラスターの参照構成(cassandra.yaml)

```
permissions_cache_max_entries: 1000
ssl_storage_port: 7001 guardrails:
  batch_size_fail_threshold_in_kb: 640
  tombstone_warn_threshold: 1000
  tombstone_failure_threshold: 100000
  batch_size_warn_threshold_in_kb: 64
  partition_size_warn_threshold_in_mb: 100
storage_port: 7000 backup_service:
  enabled: false continuous_paging:
  max_concurrent_sessions: 60
  max_session_pages: 4
  max_page_size_mb: 8
  max_local_query_time_ms: 5000
  client_timeout_sec: 600
  cancel_timeout_sec: 5
  paused_check_interval_ms: 1
batchlog_replay_throttle_in_kb: 1024 commit_failure_policy: stop
native_transport_max_frame_size_in_mb: 256
commitlog_segment_size_in_mb: 32
memtable_flush_writers: 8
client_encryption_options:
  enabled: false inter_dc_tcp_nodelay:
false column_index_cache_size_in_kb: 2
internode_authenticator: org.apache.cassandra.auth.AllowAllInternodeAuthenticator
max_value_size_in_mb: 256
authorizer: com.datastax.bdp.cassandra.auth.DseAuthorizer
listen_on_broadcast_address: false
row_cache_save_period: 0
row_cache_class_name: org.apache.cassandra.cache.OHCPProvider
native_transport_allow_older_protocols: true disk_failure_policy: stop
native_transport_port: 9042 enable_user_defined_functions_threads: true
server_encryption_options:
  internode_encryption: none dynamic_snitch_reset_interval_in_ms:
600000
compaction_throughput_mb_per_sec: 16
data_file_directories:
- /nvme1/cassandra/data
role_manager: com.datastax.bdp.cassandra.auth.DseRoleManager leaks_detection_params:
  sampling_probability: 0.01
  max_stacks_cache_size_mb: 32
  num_access_records: 0
  max_stack_depth: 30
column_index_size_in_kb: 16
windows_timer_interval: 1 rpc_keepalive:
true commitlog_total_space_in_mb: 8192
```



```
nodesync: rate_in_kb:
  1024
snapshot_before_compaction: false tracetype_query_ttl:
86400 pick_level_on_streaming: true
system_keyspaces_filtering: false
row_cache_size_in_mb: 0
tracetype_repair_ttl: 604800
native_transport_address: 192.168.1.176
auto_bootstrap: true
commitlog_sync: periodic concurrent_validations: 0
disk_optimization_strategy: ssd
counter_cache_save_period: 7200
emulate_dbaas_defaults: false
user_defined_function_warn_micros: 500
back_pressure_enabled: false
user_defined_function_fail_micros: 10000
slow_query_log_timeout_in_ms: 500 trickle_fsync:
true streaming_keep_alive_period_in_secs: 300
io_global_queue_depth: 128
write_request_timeout_in_ms: 3000
initial_token: '-9223372036854775808'
incremental_backups: false
truncate_request_timeout_in_ms: 60000
enable_scripted_user_defined_functions: false
read_request_timeout_in_ms: 5000
request_timeout_in_ms: 10000 start_native_transport:
true metadata_directory: /var/lib/cassandra/metadata
concurrent_materialized_view_builders: 2
memtable_allocation_type: offheap_objects
saved_caches_directory: /nvme2/cassandra/saved_caches
transparent_data_encryption_options:
  enabled: false chunk_length_kb:
  64
  cipher: AES/CBC/PKCS5Padding
  key_alias: testing:1
internode_compression: dc
authenticator: com.datastax.bdp.cassandra.auth.DseAuthenticator phi_convict_threshold: 8
max_hints_delivery_threads: 2
cross_node_timeout: false
partitioner: org.apache.cassandra.dht.Murmur3Partitioner
max_pending_lw_transactions: 10000 hinted_handoff_enabled:
true
hints_flush_period_in_ms: 10000 enable_user_defined_functions:
false hinted_handoff_throttle_in_kb: 1024
max_hint_window_in_ms: 10800000
otc_coalescing_strategy: DISABLED
user_defined_function_warn_heap_mb: 200
auto_snapshot: true
cdc_raw_directory: /var/lib/cassandra/cdc_raw
```




```
range_request_timeout_in_ms: 10000
user_function_timeout_policy: die
stream_throughput_outbound_megabits_per_sec: 200
commitlog_directory: /nvme2/cassandra/commitlog
sstable_preemptive_open_interval_in_mb: 50 seed_provider:
- class_name: org.apache.cassandra.locator.SimpleSeedProvider
  parameters:
    - seeds: 192.168.1.176, 192.168.3.122, 192.168.3.123
dynamic_snitch_update_interval_in_ms: 100
trickle_fsync_interval_in_kb: 10240
listen_address: 192.168.1.176
commitlog_sync_period_in_ms: 10000
cdc_enabled: false max_hints_file_size_in_mb:
128
counter_write_request_timeout_in_ms: 5000
zerocopy_streaming_enabled: true
cluster_name: CrowdStrike
memtable_space_in_mb: 512
zerocopy_max_sstables: 256
endpoint_snitch: GossipingPropertyFileSnitch cross_dc_rtt_in_ms: 0
streaming_connections_per_host: 1 hints_directory:
/var/lib/cassandra/hints
dynamic_snitch_badness_threshold: 0.1
concurrent_lw_transactions: 128
native_transport_broadcast_address: 192.168.1.176
native_transport_max_concurrent_connections_per_ip: -1
broadcast_address: 192.168.1.176
user_defined_function_fail_heap_mb: 500
permissions_validity_in_ms: 120000
roles_validity_in_ms: 120000
cas_contention_timeout_in_ms: 1000
seed_gossip_probability: 1.0
zerocopy_max_unused_metadata_in_mb: 200
native_transport_max_concurrent_connections: -1
aggregated_request_timeout_in_ms: 120000
```

正しい購入決定をする。プリセールスのスペシャリストにお問い合わせください。



Chat



Email



Call



Get updates

© Copyright 2020 Hewlett Packard Enterprise Development LP.本書の内容は、予告なしに変更されることがあります。Hewlett Packard Enterprise製品およびサービスに対する保証については、当該製品およびサービスの保証規定書に記載されています。本書のいかなる内容も、新たな保証を追加するものではありません。本書の内容につきましては万全を期しておりますが、本書中の技術的あるいは校正上の誤り、省略に対しては責任を負いかねますのでご了承ください。

Dockerは、米国およびその他の国におけるDocker, Inc.の商標または登録商標です。Intel Xeonは、米国およびその他の国におけるIntel Corporationの商標です。Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。Microsoftは、米国Microsoft Corporation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。Red Hat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。すべてのサードパーティの商標は、それぞれの所有者に帰属します。