ORACLE

OCI File Storage Performance Characteristics

2023年8月、バージョン2.0

免責事項

このドキュメントには、ソフトウェアまたは印刷物などの形式を問わず、オラクルが独占的な権利を有する財産的情報が含まれています。この機密資料へのアクセスと使用は、お客様とオラクルとの間で締結され、お客様が遵守に同意したオラクル・ソフトウェア・ライセンスおよびサービス契約の条件に従うものとします。このドキュメントとその内容の開示、コピー、複製および配布には、オラクルによる事前の承諾を必要とします。このドキュメントはライセンス契約の一部となるものではなく、オラクルおよびその子会社や関連会社との契約を構成するものではありません。

このドキュメントは情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能を提供することのコミットメント（確約）ではないため、購買決定を行う際の判断材料になさらないでください。このドキュメントに記載されている機能の開発、リリース、および時期については、オラクルの単独の裁量により決定されます。製品アーキテクチャの性質により、コードの大幅な不安定化を招くリスクを冒さずにこのドキュメントに記載されているすべての機能を安全に組み込むことは不可能な場合もあります。

改訂履歴

このドキュメントは、初版の公開後、次の改訂がありました。

日付	改訂内容
2023年8月	タイトル、内容、テスト結果を更新
2022年5月	レビューを行い、内容が最新であることを確認
2021年8月	テンプレートを更新し、軽微な編集を実施
2019年6月	軽微な更新と追加を実施
2018年9月	初版

目次

目的

Oracle Cloud Infrastructure (OCI) File Storage は、汎用のファイル共有やパフォーマンス負荷の高いワークロード(メディア処理、人工知能、機械学習など)をはじめとする様々な目的で使用できる、多用途のファイル・ストレージ・サービスです。パフォーマンス負荷の高いワークロードを File Storage にデプロイする際には、いくつかのパフォーマンス特性を理解しておくことが重要です。本書では、File Storage で最適なパフォーマンス・レベルを実現するためのベスト・プラクティスを提供し、File Storage のパフォーマンス特性と予想されるパフォーマンスについて説明するとともに、各種パフォーマンス・ベンチマークを実施した結果を示します。

アーキテクチャの概要

OCI File Storage は、フルマネージド型のスケーラブルかつセキュアなファイル・ストレージ・サービスであり、数千のコンピュート・インスタンスによる同時アクセスが可能です。あらかじめストレージをプロビジョニングしなくても、エクサバイト規模までスケールアップできます。File Storage は、Network File System (NFSv3)プロトコルを使用して、Portable Operating System Interface (POSIX)に準拠したファイル・システムへのアクセスを提供します。これにより、ユーザーとアプリケーションは、ファイル・システムがローカルにアタッチされた UNIX ファイル・システムであるかのように、ファイル・システムにアクセスできます。

ファイル・システム

ファイル・システムは、データにアクセスするための単一のネームスペースを提供します。File Storage は、可用性ドメイン内の多数のストレージ・ノードにデータが分散されている、分散型ファイル・システムです。ファイル・システムは、1 つまたは多数のマウント・ターゲットを介してエクスポートされます。

マウント・ターゲット

マウント・ターゲットとは、高度に分散されたファイル・システムへのアクセスに使用するネットワーク・エンドポイントです。コンピュート・インスタンスがファイル・システムにアクセスできるようにする IP アドレスを持ちます。ファイル・システムとマウント・ターゲットの対応は柔軟で、この後の図に示すように、1 対 1、1 対多、多対 1 のいずれかになります。

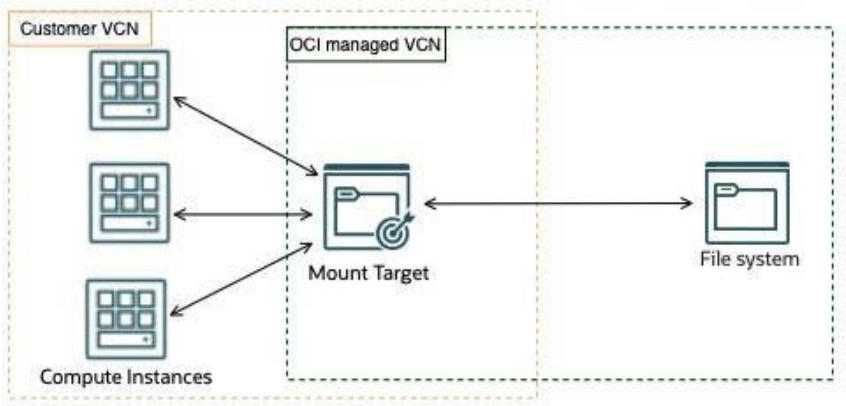

図1. マウント・ターゲットを介したファイル・システムへのアクセス - 1つのファイル・システムと1つのマウント・ターゲット

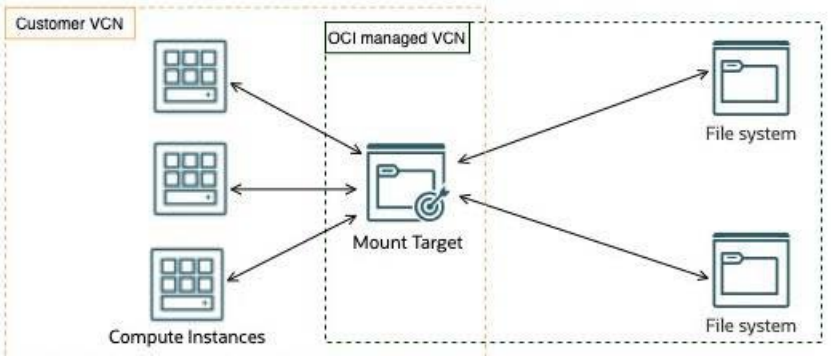

図2. マウント・ターゲットを介したファイル・システムへのアクセス - 複数のファイル・システムと1つのマウント・ターゲット

パフォーマンス特性

この項では、ワークロードを最適化して最高のパフォーマンスを確保する上で役立つように、File Storage のパフォーマンス関連の側面について詳しく説明します。

大規模なスループットとIOPS: マウント・ターゲットの水平スケーリング

File Storage のファイル・システムは、パフォーマンスの制限なしにスケールアウトできるよう設計されています。ただし、ファイル・システムにアクセスするには、コンピュート・インスタンスとファイル・システムの間にマウント・ターゲットが必要です。マウント・ターゲットは次の容量でプロビジョニングされます。

表1.マウント・ターゲットにプロビジョニングされるIOPS/スループット

タイプ	IOPS	帯域幅	クライアント数	クライアント数(TLS)	注釈
通常	50,000	1GB/秒	100,000	64	これらの数値はプロビジョニングされる容量であり、保証されるSLAではありません。

ブロック・ボリュームとは異なり、ファイル・システムへの I/O には、読取り操作と書込み操作だけでなく、CREATE、DELETE、GETATTR、SETATTR、MKDIR、RMDIR、ACCESS、RENAME などの NFSv3 メタデータ操作も含まれます。これらの操作は、File Storage バックエンドでの 1 つまたは多数の I/O 操作を伴う場合があります。たとえば、32 KiB の読取りがストレージ・バックエンドでの 2 つの I/O 操作と見なされるのに対し、GETATTR 操作はバックエンドでの 1 つの I/O 操作です。

NFS操作とIOPSの対応

次の表は、NFSv3 操作と IOPS の対応を示しています。

表2.NFS操作とIOPSの対応

NFS操作	IOPS	注釈
ACCESS 、 COMMIT 、 FSINFO 、 FSSTAT 、 GETATTR、READLINK、PATHCONF	1	軽量のNFS操作
LOOKUP、READDIR、SETATTR	2	該当なし
CREATE 、 LINK 、 MKDIR 、 MKNOD 、 READDIRPLUS、RMDIR、SYMLINK	4	該当なし
REMOVE、RENAME	8	該当なし

NFS操作	IOPS	注釈
すべてのNLM操作	4	FREE_ALLを除く
NLM FREE_ALL	8	該当なし
READ 32 KiB	2	I/Oサイズが大きく、32 KiBを超える場合は、サイズが32 KiB増えるたびに1 IOPSを追加します。 例: 読取りサイズが32 KiBを超え、64 KiB以下の場合、3 IOPSになります。
WRITE 32 KiB	4	I/Oサイズが大きく、32 KiBを超える場合は、サイズが32 KiB増えるたびに2 IOPSを追加します。

ファイル・システムの IOPS とスループットを高めるには、マウント・ターゲットを水平方向にスケーリングすると、ほぼ直線的なパフォーマンス向上を実現できます。次の図では、2 つのマウント・ターゲットを使用してファイル・システムのスループットと IOPS を 2 倍に増やしています。「予想されるパフォーマンス」の項を参照し、マウント・ターゲットをスケールアウトしてスループットと IOPS の向上を実現する方法を確認してください。

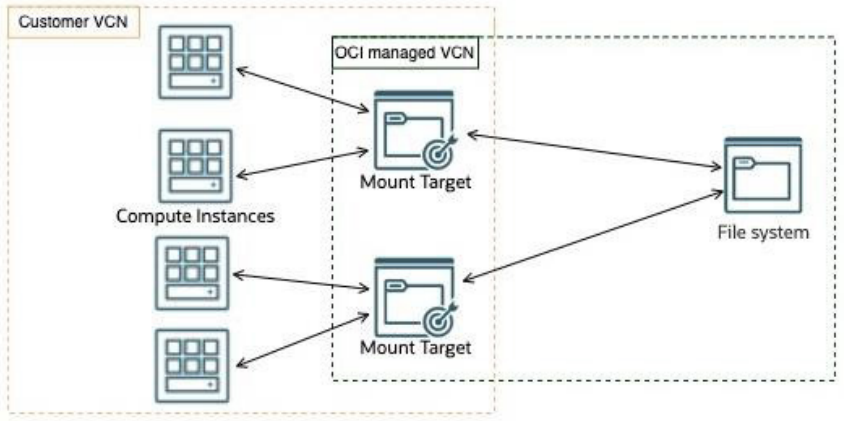

図3.マウント・ターゲットの水平スケーリング - 複数のマウント・ターゲットでファイル・システムへのスループット/IOPSを向上

マウント・ターゲット間の負荷分散

複数のマウント・ターゲット間の負荷は、均等に分散する必要があります。この分散は、コンピュート・インスタンス間で静的に行うことができます。たとえば、100 個のコンピュート・インスタンスと 4 つのマウント・ターゲットがある場合、25 個のインスタンスが 1 つのマウント・ターゲットを使用するようにします。

スループットとIOPSのモニタリング

File Storage には、File Storage をモニタリングして自動的に対策を講じるために使用できるメトリックが用意されています。

- マウント・ターゲットからのIOPSをモニタリングするには、MountTargetIOPSメトリックを使用します。
- マウント・ターゲットから得られるスループットをモニタリングするには、MountTargetReadThroughoutメトリックとMountTargetWriteThroughputメトリックを使用します。
- レイテンシをモニタリングするには、MetadataRequestAverageLatencyメトリック、FileSystemReadAverageLatencybySizeメトリック、FileSystemWriteAverageLatencybySizeメトリックを使用します。

次の図は、50K の IOPS 上限に近づいているマウント・ターゲットのモニタリング結果を示しています。

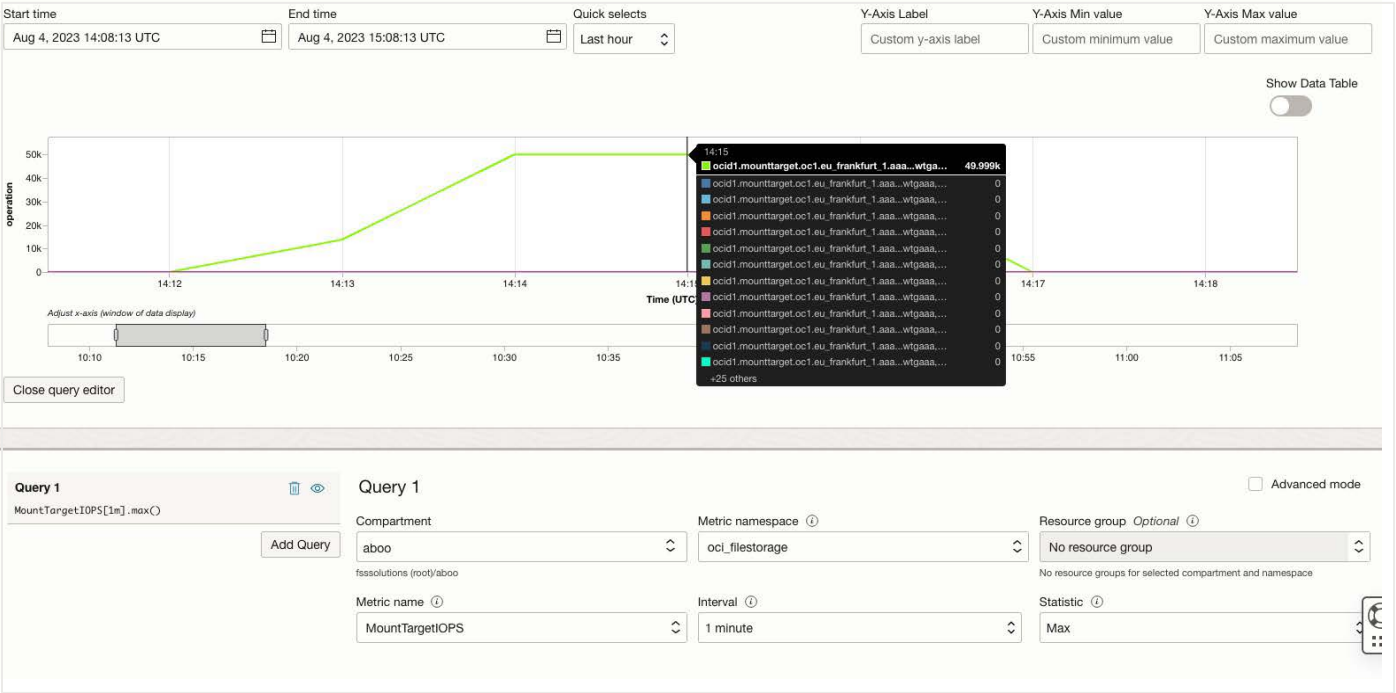

図4. 50KのIOPS上限に近づいているマウント・ターゲット

OCI Monitoring を使用すると、File Storage のメトリックをモニタリングして警告を発することができます。File Storage のモニタリング項目の 1 つに、マウント・ターゲットが最大 IOPS の 80%を超えた時点で、通知を受け取るか対策を講じるというものがあります。マウント・ターゲットが絶えずこの上限に達する場合は、マウント・ターゲットをもう 1 つ追加して負荷を分散する時期が来ています。このシナリオには次のアラーム定義を使用できます。同様に、スループット、レイテンシ、接続数などのアラームを構成できます。

基本モードのアラーム構成

アラームは、次のスクリーンショットに示すように、OCI Metrics Explorer から File Storage の様々なメトリックに基づいて構成できます。

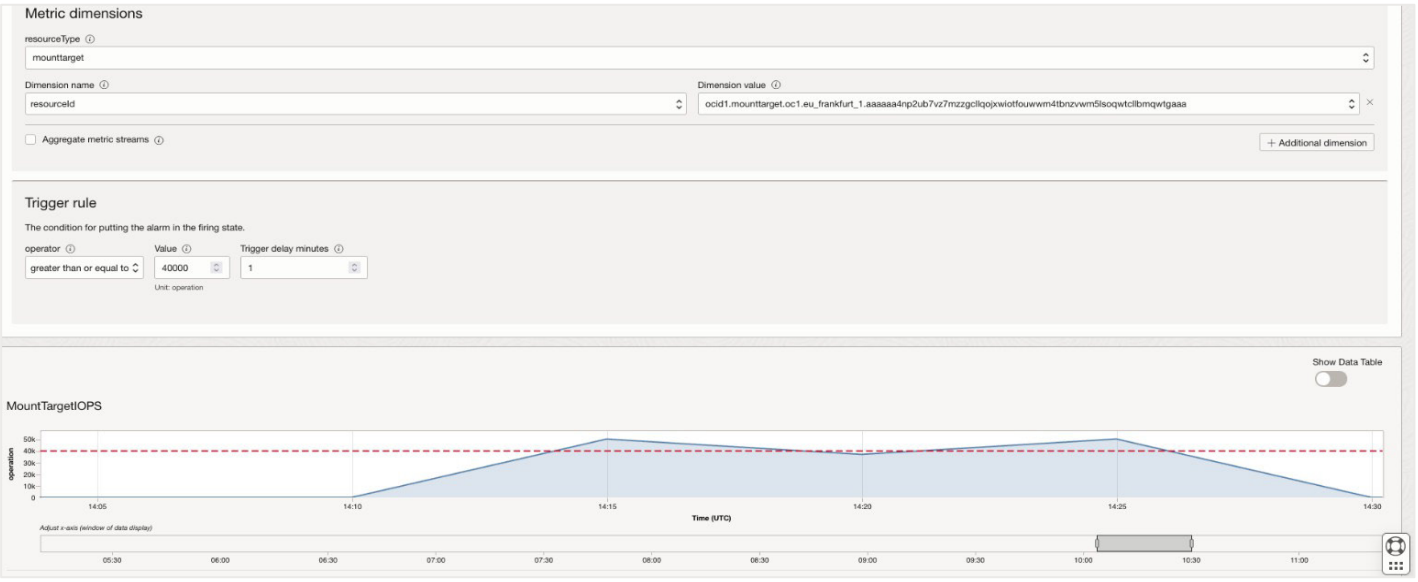

図5. 80%のIOPSのアラーム構成

拡張モードのアラーム・トリガー

次の問合せは、IOPS に基づいてアラームを設定するためのものです。

```
MountTargetIOPS[5m]{resourceType = "mounttarget", resourceId =
"ocid1.mounttarget.oc1.eu_frankfurt_1.aaaaaa4np2ub7vz7mzzgcllqojxwiotfouwwm4tbnzvwm5lsoqwtclxxxxxx"}.
max() >= 40000
```

マウント・ターゲットのサイズと制限

ワークロードに必要な IOPS とスループットを実現するには、上で説明したように、マウント・ターゲットを水平方向にスケールアウトすることをお薦めします。デフォルトでは、可用性ドメインごとに 2 つのマウント・ターゲットを使用できます。1 マウント・ターゲット当たりの最大可能スループット/IOPS にも制限があります。ただし、これらの制限はストレージの使用量に基づいて引き上げることができます。マウント・ターゲットを増やしたい、1 マウント・ターゲット当たりのスループット/IOPS をデフォルトより高くしたいなどの特定のニーズがある場合や、リージョンで予想されるパフォーマンスが得られない場合は、オラクルまでお問い合せください。ストレージの使用量とパフォーマンスの要件を評価して、お客様のテナンシに適した数とタイプのマウント・ターゲットをご提供します。

I/Oレイテンシ

I/O の制約を受けるアプリケーションの場合、考慮すべきことはスループットと IOPS だけではありません。シリアルのシングルスレッド方式で I/O 操作を実行するアプリケーションの場合は特に、I/O レイテンシもパフォーマンスの要因となる可能性があります。File Storage が提供する高度なキューの深さと並列性を利用できないため、アプリケーション・タスクが完了するまでの時間が長くなることがあります。並列化は、レイテンシを増加させることなく、FSS からワークロードの IOPS とスループットを高めるのに役立ちます。

シングルスレッドのレガシー・アプリケーション・タスクを並列化するのに役立つツールが提供されています。File Storage のパラレル・ツール(partar、parcp、parrm)を使用すると、tar、コピー、削除のワークフローでファイル操作を並列化できます。パラレル・ツールの優れた使用例の 1 つに、アプリケーションへのパッチ適用プロセスの高速化があります。このプロセスでは、多くの場合、古いファイルを削除して新しいファイルを抽出する必要がありますが、これには時間がかかります。ところが、パラレル・ツールではこうしたファイル操作を並列化できるため、アプリケーションへのパッチ適用にかかる時間が大幅に短縮されます。

次の表と図は、高度なキューの深さと同時実行性を使用することによるパフォーマンスの向上を示しています。partar ツールは、I/O 操作を並列で実行して、マウント・ターゲットへの IOPS を高めます。この例では、従来の tar で大規模な tarball からファイルを抽出するのに 39 分かかるのに対し、partar では 15 分しかかかりません。

表3.tarとpartarの完了までの時間とIOPSの違い

ツール	完了までの時間(分)	使用されたIOPS
tar	39	1300
partar	15	5100

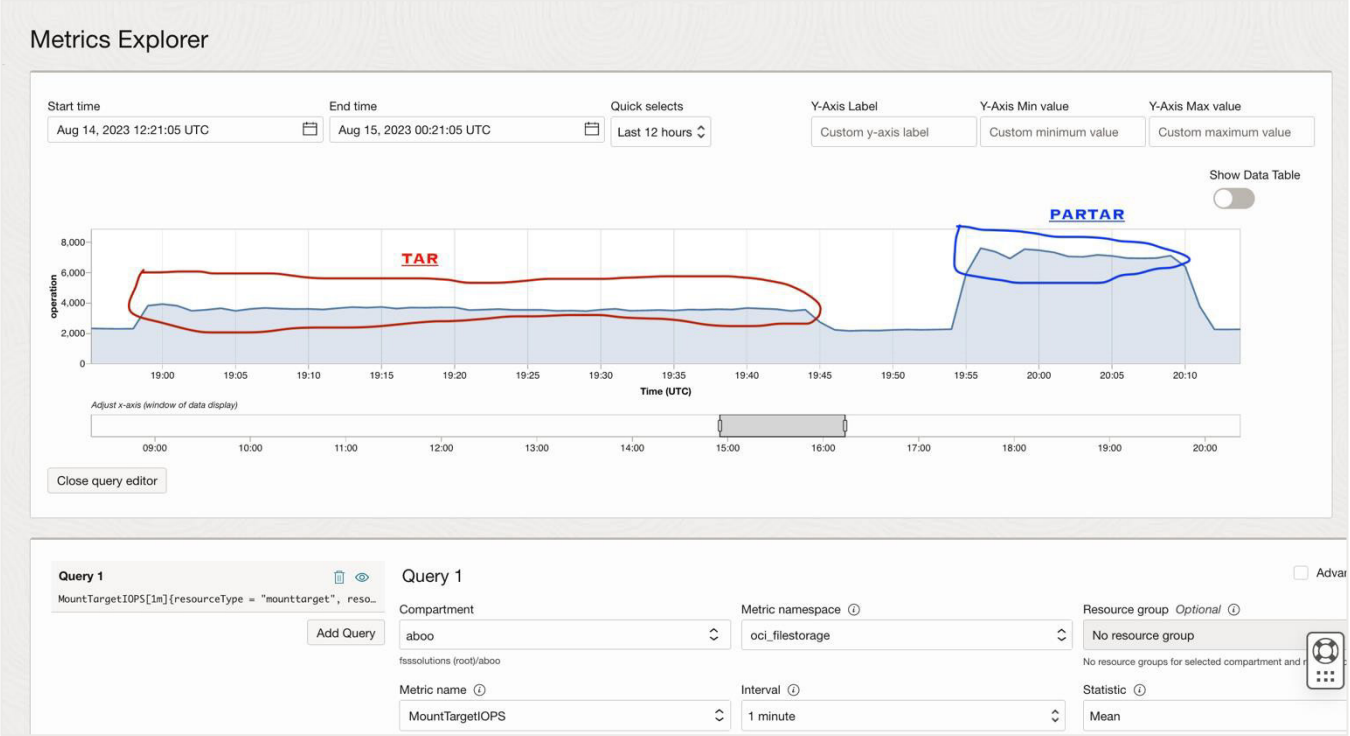

図6. tarとpartarのIOPSの違い

ネットワーク・レイテンシ

ネットワーク・レイテンシが大きくなると、I/O レイテンシも大きくなる可能性があります。TCP のウィンドウ・サイズと大きなネットワーク・レイテンシは、帯域幅遅延積により、ネットワーキング・スループットに影響を与えます。この理由から、複数の OCI リージョンやラウンド・トリップ時間(RTT)が長い複数のネットワークをまたがった Direct NFS マウントは避けてください。最適なパフォーマンスを得るには、コンピュート・インスタンスとマウント・ターゲットの両方を同じ可用性ドメインに配置することをお薦めします。

レイテンシは一般に、RTT が大きいネットワーク間での NFS および TCP 接続に関する制限です。クライアントからの TCP 接続の数を増やすと、大きい RTT の影響を軽減するのに役立ちます。1 クライアント当たりの接続数が増えると、マウント・ターゲットで I/O を提供しているスレッドの数も増えるため、スループットと IOPS が向上します。次の項の `nconnect` マウント・オプションを参照してください。

RTT が大きいネットワーク間でデータ移行またはデータ移動を行う場合は、NFS を使用して File Storage を直接クロスマウントするのではなく、インスタンス・ストリーミングを使用します。

NFSマウント・オプション

Oracle Linux 8 などの最新の Linux オペレーティング・システムでは、`nconnect=16` パラメータを使用するとパフォーマンスが向上します。このパラメータにより、Linux クライアントは最大 16 の TCP 接続を維持し、これらの TCP 接続間で I/O リクエストを多重化することができます。

最高のパフォーマンスを得るには、アプリケーションで必要とされないかぎり、ファイル・システムをマウントする際に `rsize` オプションと `wsize` オプションを設定しないでください。これらのオプションを設定しない場合は、最適な読取りサイズと書込みサイズが自動的にネゴシエートされます。読取りサイズと書込みサイズが大きければ大きいほど、スループットは高くなります。

マウント・ターゲットでのキャッシュ

マウント・ターゲットでキャッシュを行うと、NFS 操作のレイテンシが改善します。ワークロードがシングルスレッドでレイテンシの影響を受ける場合、キャッシュを行うとレイテンシが短縮されるため、パフォーマンスが大幅に向上する可能性があります。ファイル・システムが複数のマウント・ターゲットをまたがってエクスポートされないかぎり、キャッシュは常に有効です。この理由から、マウント・ターゲットの水平スケーリングでは、複数のマウント・ターゲットを介してファイル・システムをエクスポートする必要があるため、キャッシュは使用できません。

ディレクトリのサイズ

File Storage では、ファイル・システム全体のファイルの総数を数十億ファイルまでスケーリングできます。1 つのフラットなディレクトリであれば File Storage によって課される制限はありませんが、各ディレクトリを 10,000 ファイル未満に抑えることをお薦めします。ディレクトリが大きくなると、ディレクトリを頻繁に読み取る必要のあるアプリケーションの処理速度が低下する可能性があります。これは、File Storage のアーキテクチャ上の制限ではなく、大規模なディレクトリに NFS 経由でアクセスする場合の一般的な状況です。

インスタンスの容量

インスタンスの使用可能なネットワーク帯域幅は I/O のパフォーマンスに影響を与えます。OCI では、サイズの大きいインスタンス(CPU が多い)が、より広いネットワーク帯域幅を使用する権限を持ちます。File Storage のパフォーマンスが最大になるのは、OCI Compute ベア・メタル・インスタンスまたはサイズの大きい VM シェイプを使用する場合です。

予想されるパフォーマンス

最も高いパフォーマンス・レベルは同時アクセスを前提としており、複数のクライアント、複数のスレッドおよび複数のマウント・ターゲットを使用することでのみ達成可能です。1 つのマウント・ターゲットを使用して達成できる実際のスループットと IOPS は、I/O のタイプとサイズ、インスタンスの容量、I/O のパターンなど、多くの要因に左右されます。この理由から、本書では、マウント・ターゲットのスケーリングの実例を示し、読者がワークロードの IOPS とスループットの予想を立てられるようにサポートするにあたり、実践的なアプローチを取っています。

テストは、マウント・ターゲットの水平スケーリングによって直線的に拡大できることを示すために、1 つのファイル・システムにアタッチされた最大 8 つのマウント・ターゲットを使用して実施されました。この項に記載されている IOPS は読取りまたは書込みの NFS 操作です。これらは、マウント・ターゲットでの IOPS に変換できます。この計算には、「NFS 操作と IOPS の対応」の項で説明した読取りサイズと書込みサイズに対応する IOPS を使用します。

テスト環境

リージョン: eu-frankfurt-1/AD3
ファイル・システムのデータセット・サイズ: 4 TiB
マウント・ターゲット: 同じファイル・システムをエクスポートするマウント・ターゲットが 8 つあり、各インスタンスに 1 つのマウント・ターゲットが対応している
IO タイプ: ランダムな読取りと書込み

インスタンス	シェイプ	サイズ	メモリー	ネットワーク	OS
OCI VM x 8	VM.Standard2	16	240 GB	16 Gbps	Oracle Linux 8

テスト方法: FIO

```
$ fio --name=read_throughput --directory=/fss --numjobs=8 -size=8G --time_based --runtime=180 --
ioengine=libaio --direct=1 --verify=0 --bs=$IOSIZE --iodepth=32 --rw=randread --group_reporting=1

$ fio --name=write_throughput --directory=/fss --numjobs=8 -size=8G --time_based --runtime=180 --
ioengine=libaio --direct=1 --verify=0 --bs=$IOSIZE --iodepth=32 --rw=randwrite --group_reporting=1
```

読取りのテスト結果: IOPSの最適化

I/Oサイズ	マウント・ターゲットの数	1マウント・ターゲット当たりの平均読取りIOPS	FSの合計読取りIOPS	1マウント・ターゲット当たりの平均スループット(MB/秒)	FSの合計スループット(MB/秒)
32 KiB	1	25018	25018	819	819
32 KiB	2	25057	50115	820	1640
32 KiB	4	25056	100226	818	3279
32 KiB	8	24444	195554	800	6402

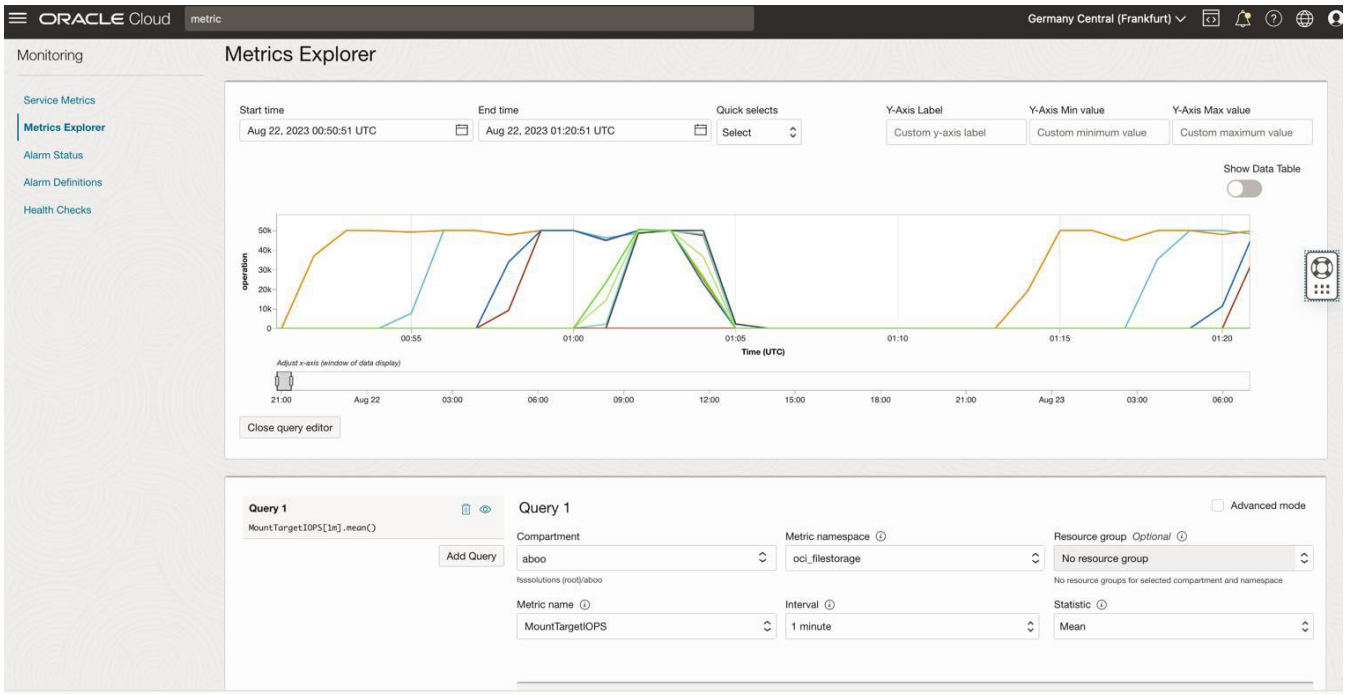

図7. 32 KiBのIOサイズで1～8個のマウント・ターゲットを使用した読取り

読取りのテスト結果: I/Oサイズが大きい場合のスループットの最適化

I/Oサイズ	マウント・ターゲットの数	1マウント・ターゲット当たりの平均読取りIOPS	FSの合計読取りIOPS	1マウント・ターゲット当たりの平均スループット(MB/秒)	FSの合計スループット(MB/秒)
1 MiB	1	937	937	983	983
1 MiB	2	964	1929	1008	2016
1 MiB	4	955	3923	990	3960
1 MiB	8	949	7594	990	7918

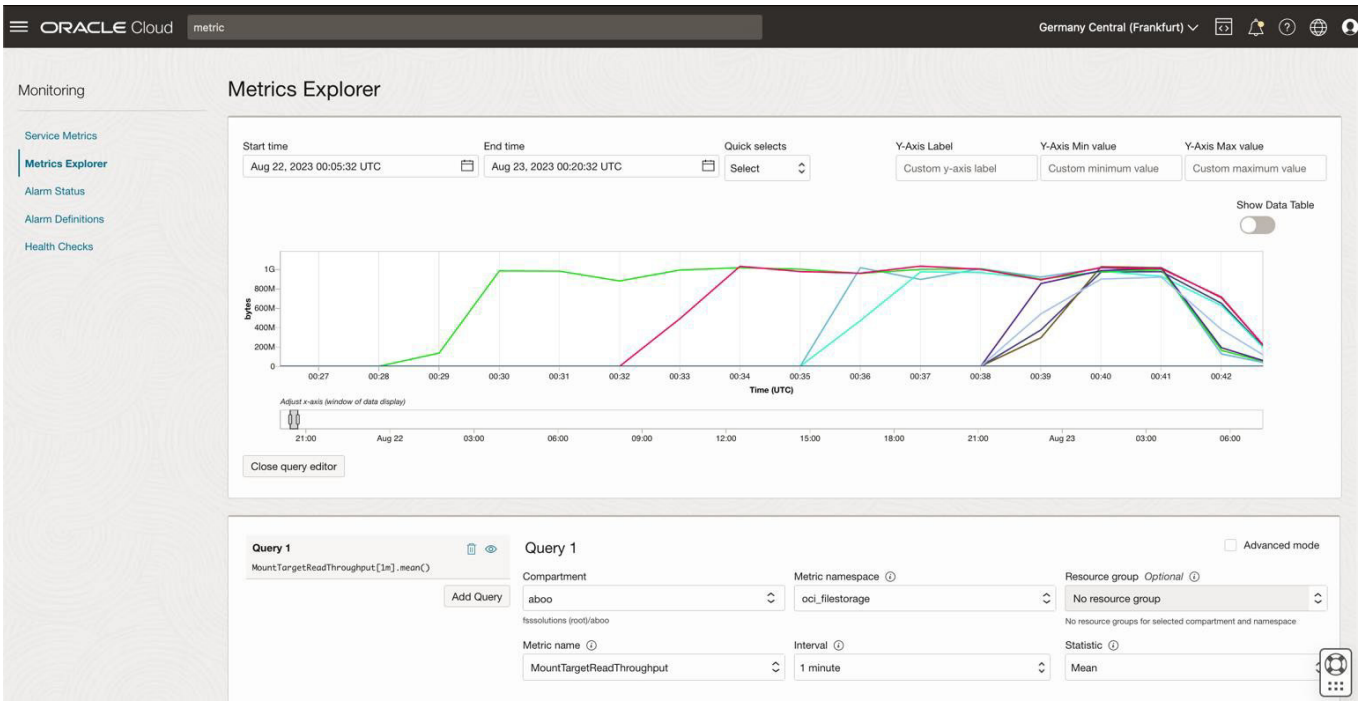

図8. 1 MiBのIOサイズで1～8個のマウント・ターゲットを使用した読取り

書込みのテスト結果: IOPSの最適化

I/Oサイズ	マウント・ターゲットの数	1マウント・ターゲット当たりの平均書込みIOPS	FSの合計書込みIOPS	1マウント・ターゲット当たりの平均スループット(MB/秒)	FSの合計スループット(MB/秒)
32 KiB	1	12580	12580	412	412
32 KiB	2	12575	25151	412	824
32 KiB	4	12580	50321	412	1648
32 KiB	8	12574	100599	411	3295

書込みのテスト結果: I/Oサイズが大きい場合のスループットの最適化

I/Oサイズ	マウントの数	1マウント・ターゲット当たりの平均書込みIOPS	FSの合計書込みIOPS	1マウント・ターゲット当たりの平均スループット(MB/秒)	FSの合計スループット(MB/秒)
1 MiB	1	761	761	799	799
1 MiB	2	761	1523	799	1598
1 MiB	4	761	3046	799	3196
1 MiB	8	761	6093	799	6392

読取りと書込みが混在するシナリオ

読取りと書込みが混在する操作のパフォーマンスも、順次かランダムかに関係なく同様です。前述のとおり、NFS の読取り操作と書込み操作は、I/O のサイズに基づいたマウント・ターゲットの IOPS に変換できます。達成可能なパフォーマンスは、マウント・ターゲットで使用できる IOPS と帯域幅によって制限されます。

結論

OCI File Storage のファイル・システムは、ストレージのサイズやパフォーマンスをあらかじめプロビジョニングしなくても、要件に合せてスケーリングできます。File Storage は、高い IOPS とスループットを必要とするワークロードに使用できます。実例を示したように、複数のマウント・ターゲットにスケールアウトすることで、必要なパフォーマンス・レベルを達成できます。ご質問や特定のパフォーマンス要件がある場合は、電子メールで、または OCI のヘルプとサポートからお問い合せください。