テクニカルレポート
How to enable StorageGRID in your environment

NetApp
April 26, 2024

目次

テクニカルレポート

NetApp StorageGRIDとビッグデータ分析

NetApp StorageGRIDのユースケース

NetApp StorageGRIDオブジェクトストレージ解決策は、拡張性、データ可用性、セキュリティ、ハイパフォーマンスを提供します。StorageGRID S3は、あらゆる規模のさまざまな業界の組織で幅広いユースケースに使用されています。典型的なシナリオをいくつか見てみましょう。

ビッグデータ分析： StorageGRID S3はデータレイクとしてよく使用されています。企業は、Apache Spark 、Splunk Smartstore、Dremioなどのツールを使用して、分析用に大量の構造化データと非構造化データを保存します。

データ階層化： NetAppのお客様は、ONTAPのFabricPool機能を使用して、ハイパフォーマンスなローカル階層間でStorageGRIDにデータを自動的に移動します。階層化することで、高価なフラッシュストレージをホットデータ用に解放し、コールドデータを低コストのオブジェクトストレージでいつでも利用できる状態に維持できます。これにより、パフォーマンスとコスト削減が最大化されます。

*データのバックアップとディザスタリカバリ：*企業は、StorageGRID S3を信頼性とコスト効率に優れた解決策として使用して、重要なデータのバックアップと災害時のリカバリを実行できます。

アプリケーション用のデータストレージ： StorageGRID S3はアプリケーションのストレージバックエンドとして使用できるため、開発者はファイル、画像、ビデオ、その他の種類のデータを簡単に保存および取得できます。

コンテンツ配信： StorageGRID S3を使用すると、静的なWebサイトコンテンツ、メディアファイル、ソフトウェアダウンロードを世界中のユーザに保存して配信できます。StorageGRIDの地理的な配信とグローバルネームスペースを活用して、高速で信頼性の高いコンテンツ配信を実現できます。

データ階層化： NetAppのお客様は、ONTAP FabricPool機能を使用して、ハイパフォーマンスなローカル階層間でStorageGRIDにデータを自動的に移動します。階層化することで、高価なフラッシュストレージをホットデータ用に解放し、コールドデータを低コストのオブジェクトストレージからいつでも利用できる状態に保ちます。これにより、パフォーマンスとコスト削減が最大化されます。

データアーカイブ： StorageGRIDは、さまざまな種類のストレージを提供し、パブリックな長期低コストストレージオプションへの階層化をサポートします。コンプライアンスや履歴目的で保持する必要があるデータのアーカイブや長期保存に最適な解決策です。

オブジェクトストレージのユースケース

[StorageGRIDのユースケース図、幅= 396、高さ= 394]

上記の中で、ビッグデータ分析は最も多くのユースケースの1つであり、その使用量は増加傾向にあります。

データレイクにStorageGRIDを選ぶ理由

- コラボレーションの強化-業界標準のAPIアクセスによる大規模な共有マルチサイト、マルチテナンシー
- 運用コストの削減-単一の自己回復型自動スケールアウトアーキテクチャによる運用の簡易化
- 拡張性-従来のHadoopやデータウェアハウスソリューションとは異なり、StorageGRID S3オブジェクトス

トレージはコンピューティングやデータからストレージを切り離し、ビジネスの成長に合わせてストレージニーズを拡張できます。

- 耐久性と信頼性- StorageGRIDは99.999999999%の耐久性を提供し、保存されたデータはデータ損失に対して非常に耐性があります。また、高可用性を提供し、データへの常時アクセスを保証します。
- セキュリティ- StorageGRIDは、暗号化、アクセス制御ポリシー、データライフサイクル管理、オブジェクトロック、S3バケットに格納されたデータを保護するバージョン管理など、さまざまなセキュリティ機能を提供します。
- StorageGRID S3データレイク*

[StorageGRIDデータレイクの例、幅= 614、高さ= 345]

S3オブジェクトストレージに最も適したデータウェアハウスまたはデータレイク

NetAppは、Hive、Delta Lake、Dremioの3つのデータウェアハウス/レイクハウスエコシステムでStorageGRIDをベンチマークしました。 "『Apache Iceberg: The Definitive Guide』" データウェアハウスとデータレイクハウスの簡単な紹介と、これら2つのアーキテクチャの長所と短所が含まれています。

- ベンチマークツール- TPC-DS- https://www.tpc.org/tpcds/
- ビッグデータエコシステム
  - 5台のVMで構成されるクラスタ。各VMに128G RAM、24個のvCPU、システムディスク用SSDストレージが搭載されています。
  - Hadoop 3.3.5とHive 3.1.3（1つのネームノード+ 4つのデータノード）
  - Delta LakeとSpark 3.2.0（1マスター+ 4ワーカー）およびHadoop 3.3.5
  - Dremio v23（マスター1名+エグゼキューター4名）
- オブジェクトストレージ
  - SG6060を3台+ SG1000ロードバランサを1台搭載した場合、NetApp ^ StorageGRID®SG®^11.6
  - オブジェクトの保護-コピー×2
- データベースサイズ1000GB
- クエリテストごとに一貫した結果を得るために、3つのエコシステムすべてでキャッシュが無効になりました。

TPC-DSには、クエリベンチマーク用に99の複雑なSQLクエリが付属しています。99個のクエリをすべて完了するまでの合計時間を分単位で測定し、結果を分析するためにS3要求のタイプと数を細かく分析しました。次の表は、全99件のクエリの合計期間を示しています。2番目の表は、各エコシステムがStorageGRIDに送信するS3要求の数とタイプを示しています。

- TPC-DSクエリ結果*

エコシステム	ハイブ	デルタレイク"	デレミオ
ストレージレイヤ	NetApp ® StorageGRID ®	NetApp ® StorageGRID ®	NetApp ® StorageGRID ®
ドライブタイプ	HDD	HDD	HDD
表形式	寄木細工	寄木細工	寄木細工[1]

エコシステム	ハイブ	デルタレイク"	デレミオ
データベースサイズ	1000 g	1000 g	1000 g
TPCDS 99クエリ+ 合計分数	1084[2]	55	47です

[1]寄木細工と氷山の両方のテーブル形式をテストしましたが、結果は似ています。

[2]Hiveクエリー番号72を完了できません。

- TPC-DSクエリ- S3要求の内訳*

S3要求	ハイブ	デルタレイク"	デレミオ
取得	一、一一七、一八四	2、074、610	四、四一四、二二二七
観察：+ すべての範囲GET	80%のGET：32MBオブジェクトから2KB～2MB、50~100要求/秒	73%の範囲は、32MBオブジェクトから100KB未満、1、000~1400要求/秒	90% 1Mバイト範囲は256MBのオブジェクトから取得、2000~2300の要求/秒
オブジェクトをリスト表示	三一二、〇 五三	二四、一五八	240
頭部+ （存在しないオブジェクト）	156、027	一二、一 〇 三	192年
頭部+ （既存のオブジェクト）	982、126	922、732	一、八四五
リクエスト総数	二、五六七、三九 〇	3、033、603	4、416、504

最初のテーブルから、デルタ湖とDremioがHiveよりもはるかに速いことがわかります。2つ目の表から、Hiveが大量のS3リストオブジェクト要求を送信していることがわかります。この要求は、すべてのオブジェクトストレージプラットフォーム（特に多数のオブジェクトを含むバケットを扱う場合）では通常低速です。これにより、全体的なクエリ時間が大幅に長くなります。もう1つの観測点は、Dremioが大量のGET要求を並行して送信することができたことで、Hiveでは毎秒50~100件の要求に対して、毎秒2,000～2300件の要求が送信されたことです。HiveとHadoop S3AのMimic standard filesystemは、S3オブジェクトストレージのHiveの低速化に貢献しています。

Hadoop（HDFSまたはS3オブジェクトストレージ上）をHiveまたはSparkで使用するには、HadoopとHive/Sparkの広範な知識と、各サービスの設定の相互作用に関する知識が必要です。これらの設定の合計数は1000以上です。多くの場合、設定は相互に関連しており、単独で変更することはできません。使用する設定と値の最適な組み合わせを見つけるには、膨大な時間と労力がかかります。

Dremioは、エンドツーエンドのApache Arrowを使用してクエリのパフォーマンスを劇的に向上させるデータレイクエンジンです。Apache Arrowは、効率的なデータ共有と高速分析のために標準化されたカラムナメモリフォーマットを提供します。Arrowは、言語に依存しないアプローチを採用しており、データのシリアライゼーションとデシリアライゼーションの必要性を排除し、複雑なデータプロセスとシステム間のパフォーマンスと相互運用性を向上させるように設計されています。

Dremioの性能は主にDremioクラスター上の計算能力によって駆動される。DremioはS3オブジェクトストレージ接続にHadoopのS3Aコネクタを使用しますが、Hadoopは必須ではなく、Hadoopのfs.s3a設定のほとんどはDremioでは使用されません。これにより、さまざまなHadoop s3a設定の学習とテストに時間を費やすことなく、Dremioのパフォーマンスを簡単に調整できます。

このベンチマーク結果から、S3ベースのワークロード向けに最適化されたビッグデータ分析システムがパフォーマンスの大きな要因であることがわかります。Dremioはクエリの実行を最適化し、メタデータを効率的に利用し、S3データへのシームレスなアクセスを提供するため、S3ストレージを使用する場合にHiveと比較してパフォーマンスが向上します。これを参照してください "ページ" StorageGRIDでDremio S3データソースを設定するには、次の手順を実行します。

StorageGRIDとDremioが連携して最新の効率的なデータレイクインフラを提供する方法や、NetAppがHive + HDFSからDremio + StorageGRIDに移行してビッグデータ分析の効率を劇的に向上させる方法については、以下のリンクをご覧ください。

- "NetApp StorageGRIDでビッグデータのパフォーマンスを向上"
- "StorageGRIDとDremioによる、パワフルで効率性に優れた最新のデータレイクインフラ"
- "NetAppが製品分析でカスタマーエクスペリエンスを再定義する方法"

Hadoop S3Aの調整

Hadoop S3Aコネクタは、HadoopベースのアプリケーションとS3オブジェクトストレージ間のシームレスなやり取りを容易にします。S3オブジェクトストレージを使用する際のパフォーマンスを最適化するには、Hadoop S3Aコネクタの調整が不可欠です。調整の詳細に進む前に、Hadoopとそのコンポーネントの基本を理解しておきましょう。

Hadoopとは

- Hadoop * は、大規模なデータ処理とストレージを処理するために設計された強力なオープンソース・フレームワークです。これにより、コンピュータのクラスタ間で分散ストレージと並列処理が可能になります。

Hadoopの3つのコアコンポーネントは次のとおりです。

- * Hadoop HDFS（Hadoop分散ファイルシステム）*：ストレージを処理し、データをブロックに分割してノード間で分散します。
- * Hadoop MapReduce *：タスクを小さなチャンクに分割し、並行して実行することでデータを処理します。
- * Hadoop YARN（Yet Another Resource Negotiator）：* "リソースの管理とタスクのスケジュール設定を効率的に行う"

Hadoop HDFSおよびS3Aコネクタ

HDFSはHadoopエコシステムの重要なコンポーネントであり、効率的なビッグデータ処理において重要な役割を果たします。HDFSは信頼性の高いストレージと管理を実現します。並列処理と最適化されたデータストレージを実現し、データアクセスと分析を高速化します。

ビッグデータ処理では、HDFSは大規模データセットにフォールトトレラントなストレージを提供することに優れています。これは、データレプリケーションによって実現されます。IT部門は、データウェアハウス環境に大量の構造化データと非構造化データを格納して管理できます。さらに、Apache Spark、Hive、Pig、Flinkなどの主要なビッグデータ処理フレームワークとシームレスに統合し、スケーラブルで効率的なデータ処理を可能にします。UNIXベース(Linux)オペレーティングシステムと互換性があり、ビッグデータ処理にLinuxベースの環境を使用することを好む組織にとって理想的な選択肢です。

時間の経過とともにデータ量が増大するにつれて、独自のコンピューティングとストレージを使用し

てHadoopクラスタに新しいマシンを追加するアプローチは非効率的になります。リニアに拡張すると、リソースの効率的な使用やインフラの管理が難しくなります。

これらの課題に対処するために、Hadoop S3AコネクタはS3オブジェクトストレージに対するハイパフォーマンスI/Oを提供します。S3Aを使用してHadoopワークフローを実装することで、オブジェクトストレージをデータリポジトリとして活用でき、コンピューティングとストレージを分離することができます。これにより、コンピューティングとストレージを別々に拡張できます。コンピューティングとストレージを分離することで、コンピューティングジョブ専用のリソースを確保し、データセットのサイズに基づいて容量を提供することもできます。そのため、Hadoopワークフローの総所有コストを削減することができます。

Hadoop S3Aコネクタの調整

S3の動作はHDFSとは異なり、ファイルシステムの外観を維持しようとすると積極的に最適化されません。S3リソースを最も効率的に使用するには、慎重な調整、テスト、実験が必要です。

本ドキュメントのHadoopオプションはHadoop 3.3.5に基づいています。を参照してください。 "Hadoop 3.3.5 core-site.xml" 使用可能なすべてのオプションについて。

注–一部のHadoop fs.s3a設定のデフォルト値は、Hadoopのバージョンによって異なります。現在のHadoopバージョンに固有のデフォルト値を確認してください。これらの設定がHadoop core-site.xmlに指定されていない場合は、デフォルト値が使用されます。SparkまたはHive構成オプションを使用して、実行時に値を上書きできます。

これに行く必要があります。 "Apache Hadoopページ" 各fs.s3aオプションを理解するため。可能であれば、非本番環境のHadoopクラスタでテストして最適な値を特定します。

お読みください "S3Aコネクタでの作業時のパフォーマンスの最大化" その他のチューニングの推奨事項については、

主な考慮事項をいくつか見ていきましょう。

- 1 。データ圧縮*

StorageGRID圧縮を有効にしないでください。ほとんどのビッグデータシステムでは、オブジェクト全体を読み出す代わりにバイト範囲GETを使用します。圧縮オブジェクトにbyte range getを使用すると、GETのパフォーマンスが大幅に低下します。

※ 2S3Aコミッタ*

一般的には、マジックs3aコミッターをお勧めします。これを参照してください "共通のS3Aコミッタオプションページ" マジックコミッタとそれに関連するs3a設定をよりよく理解するため。

マジックコミッター：

Magic Committerは、特にS3Guardを使用して、S3オブジェクトストアで一貫したディレクトリリストを提供します。

整合性のあるS3（現在はそうなっています）を使用すると、Magic Committerは任意のS3バケットで安全に使用できます。

選択と実験：

ユースケースに応じて、Staging Committer（クラスタHDFSファイルシステムに依存）とMagic Committerの

どちらかを選択できます。

両方を試して、ワークロードと要件に最適なものを判断してください。

要約すると、S3Aコミッタは、S3への一貫した、高性能で信頼性の高い出力コミットメントという基本的な課題に対する解決策を提供します。内部設計により、データの整合性を維持しながら効率的なデータ転送を実現します。

[S3Aオプションテーブル]

3.スレッド、接続プールサイズ、ブロックサイズ

- 1つのバケットとやり取りする各* S3A *クライアントには、アップロードおよびコピー処理用のオープンHTTP 1.1接続とスレッドの専用プールがあります。
- "これらのプールサイズを調整して、パフォーマンスとメモリ/スレッド使用量のバランスをとることができます。"。
- S3にデータをアップロードする場合、データはブロックに分割されます。デフォルトのブロックサイズは32MBです。この値をカスタマイズするには、fs.s3a.block.sizeプロパティを設定します。
- ブロックサイズを大きくすると、アップロード中にマルチパートパートパートを管理するオーバーヘッドが軽減されるため、大規模なデータアップロードのパフォーマンスが向上します。大規模なデータセットの場合、推奨値は256 MB以上です。

[S3Aオプションテーブル]

4.マルチパートアップロード

s3aコミッタ*常に* MPU（マルチパートアップロード）を使用してデータをs3バケットにアップロードします。これは、タスクの失敗、タスクの投機的な実行、およびコミット前のジョブの中止を可能にするために必要です。マルチパートアップロードに関連する主な仕様を次に示します。

- 最大オブジェクトサイズ：5TiB（テラバイト）。
- アップロードあたりの最大パーツ数：10、000
- パーツ番号：1～10,000（含む）。
- パーツサイズ：5MiB~5GiB。特に、マルチパートアップロードの最後のパートには最小サイズの制限はありません。

S3マルチパートアップロードに小さいパートサイズを使用すると、メリットとデメリットの両方があります。

利点：

- ネットワークの問題からのクイックリカバリ:小さなパーツをアップロードすると、ネットワークエラーによるアップロードの再開による影響が最小限に抑えられます。パーツに障害が発生した場合は、オブジェクト全体ではなく、その特定のパーツのみを再アップロードする必要があります。
- 並列化の向上:マルチスレッディングまたは同時接続を利用して、より多くのパーツを並行してアップロードできます。この並列化により、特に大きなファイルを処理する場合のパフォーマンスが向上します。

欠点：

- ネットワークオーバーヘッド:部品サイズが小さいほど、アップロードする部品が増えます。各部品には独

自のHTTPリクエストが必要です。HTTP要求が増えると、個 々 の要求の開始と完了のオーバーヘッドが増加します。多数の小さなパーツを管理すると、パフォーマンスに影響を与える可能性があります。

- 複雑さ：注文の管理、パーツの追跡、アップロードの成功の確認は面倒です。アップロードを中止する必要がある場合は、すでにアップロードされているすべてのパーツを追跡してパージする必要があります。

Hadoopの場合、fs.s3a.multipart.sizeには256MB以上のパーツサイズを推奨します。fs.s3a.mutlipart.threshold値は常に2 x fs.s3a.multipart.size値に設定します。たとえば、fs.s3a.multipart.size=256Mの場合、fs.s3a.mutlipart.thresholdは512Mにする必要があります。

大きなデータセットには大きなパーツサイズを使用してください。特定のユースケースとネットワーク条件に基づいて、これらの要因のバランスを取る部品サイズを選択することが重要です。

マルチパートアップロードは "3段階のプロセス"：

1. アップロードが開始され、StorageGRIDはupload-idを返します。
2. オブジェクトパーツはupload-idを使用してアップロードされます。
3. すべてのオブジェクトパートがアップロードされると、は、upload-idを指定して完全なマルチパートアップロード要求を送信します。StorageGRIDは、アップロードされたパーツからオブジェクトを構築し、クライアントがオブジェクトにアクセスできるようにします。

Complete multipart upload要求が正常に送信されなかった場合、パーツはStorageGRIDに残り、オブジェクトは作成されません。これは、ジョブが中断、失敗、または中止された場合に発生します。マルチパートアップロードが完了するか中止されるか、アップロードが開始されてから15日が経過するとStorageGRIDがそれらのパートをパージするまで、パートはグリッドに残ります。バケット内で実行中のマルチパートアップロードが多数（数十万から数百万）ある場合、Hadoopが「list-multipart-uploads」を送信すると（この要求はアップロードIDでフィルタリングされません）、要求の完了に時間がかかるか、最終的にタイムアウトになることがあります。fs.s3a.mutlipart.purgeをtrueに設定し、適切なfs.s3a.multipart.purge.ageの値を設定することを検討してください（例：5〜7日、デフォルト値の86400、つまり1日は使用しないでください）。または、NetAppサポートに状況を調査してください。

[S3Aオプションテーブル]

5.メモリ内のバッファ書き込みデータ

パフォーマンスを向上させるには、書き込みデータをS3にアップロードする前にメモリにバッファします。これにより、少量の書き込み数が削減され、効率が向上します。

[S3Aオプションテーブル]

S3とHDFSは別 々 の方法で機能することに注意してください。S3リソースを最も効率的に使用するには、慎重な調整/テスト/実験が必要です。

著作権に関する情報

商標に関する情報