ORACLE
OPTIMIZED SOLUTIONS

Oracleテクニカル・ホワイト・ペーパー
2012年3月

Sun ZFS Storage Applianceにおける ファイバ・チャネルの使用について

概要 3
はじめに 4
FC LUNでのSun ZFSSAの使用 5
Sun ZFSSAファイバ・チャネル・アーキテクチャ 5
運用の理論 6
ファイバ・チャネルの設定と構成の手順 6
ファイバ・チャネルLUNを利用するための設定プロセス 7
Sun ZFSSAハードウェアの準備 7
ファイバ・チャネルのホスト接続 9
SANの構成とゾーニングの設定 10
Sun ZFSSAノード上でのLUNの作成と構成 14
ホスト上の構成の確認 16
LUNのパーティション化とラベル付け 16
SANゾーニング構成の確認 17
書込みキャッシュに関する考慮事項 19
障害およびリカバリ・シナリオの構成のテスト 21
ファブリックとホストHBAポート間のシングル接続障害 21
ファブリックとSun ZFSSAノード間のシングル接続障害 22
Sun ZFSSAノードの両方のアクティブ・ポートへのリンク障害 24
Sun ZFSSAノードの障害 24
障害シナリオのサマリー 24
設計のベスト・プラクティス 25
ホスト接続に関する考慮事項 25
コマンド・キュー深度の設定 25
OpenSolarisでのコマンド・キュー深度の設定 26
Solaris Kernelで`s(s)d_max_throttle`を設定するためのグローバル・メソッド 26

パーティション・アライメント....26
Oracle Solarisのパーティション化....28
Windows OSのパーティション化....30
Windows Server 2008とWindows Vista....30
Windows Server 2003以前....30
VMwareのパーティション化....31
VMware ESX 3.0....31
VMware ESX 4.0とESXi 4.0....31
VMware VMFSブロック・サイズ....32
Linuxのパーティション化....32
SSDキャッシュ・タイプのガイドライン....34
参照テストの設定....34
テスト結果の解釈....36
Sun ZFSSAの書込みキャッシュの使用....36
SSDキャッシュ・デバイス・タイプの比較....37
OLTPタイプのワークロード....39
結論....39
付録A：vdbenchパラメータ・ファイル....42
付録B：VMware ALUAサポートの設定....43
現在のSATPプラグイン・ルールの確認....43
ベンダーおよびモデルのident情報の判別....44
SATP構成ルールの追加と確認....44
Sun ZFSSA FC LUN ALUAパス・ステータスの確認....44
付録C：参考資料....48
参考文献....48
ブログ....48

概要

Sun ZFS Storage Appliance（ZFSSA）7000ファミリーは、ファイバ・チャネル・ターゲットLUN機能を搭載して、ユーザーがアプライアンスのストレージ・プールのボリュームを構成し、ファイバ・チャネルのブロック・デバイスとして使用できるようにします。これらのLUNを適切に構成することが、そのパフォーマンスと効率性を最大限に引き出す鍵となります。本書では、次に示すFC LUN機能を確認する方法の手順と推奨事項を説明し、いくつかの例を挙げます。

- Sun ZFSSAの2ノード・クラスタ・ストレージ・サブシステムを使用して、ファイバ・チャネル構成を設定する手順
- SANゾーニングを使用してSANで冗長I/Oパスを設計する場合、各種オペレーティング・システムに対しパーティションとファイル・システムを作成する場合の推奨事項
- 作成したFC LUNのブロック・サイズに合わせてパーティションを位置合わせする（重要機能）場合のガイドライン、推奨事項、および例
- 最後に、各種構成で実施されたオンライン・トランザクション処理（OLTP）のパフォーマンス・テストに基づいて、SSDキャッシュ・デバイスを使用する場合の推奨事項

はじめに

Sun ZFS Storage Applianceシリーズのファームウェア拡張機能により、Sun ZFSSAが顧客のSAN環境でファイバ・チャネル・ブロック・デバイスとして機能するように、ファイバ・チャネル・ターゲット・デバイスを指定することができます。

ファイバ・チャネル・ターゲット・モードでは、ホストおよびSun ZFSSA上のターゲット・デバイス間で複数のI/Oパスがサポートされます。Sun ZFSSA構成でデュアルまたはシングル・ファブリックSANのいずれかのアーキテクチャを適切に開発することは、パスの冗長性が確実に確立されることを意味します。

構成プロセスの次のステップでは、スペース、信頼性、パフォーマンス、およびアプリケーションI/Oプロファイルに関する顧客とアプリケーションの要件を、ボリューム/LUNサイズ要件、プール・プロファイルの選択肢、およびSSDタイプ・キャッシュ・デバイスの選択肢にマッピングします。本書では、設計のベスト・プラクティス、パフォーマンス関連のガイドライン、このプロセスの例について説明します。

最適なパフォーマンス結果を得られるように、構成済みのLUNにパーティションを作成することも等しく重要です。各オペレーティング・システムでは、その独自のツールを使用し、パーティションの開始場所にはデフォルト値を使用します。ただし、ほとんどの場合、デフォルト値は最適ではありません。各主要タイプのオペレーティング・システムのパーティション・ツールについて検証してから、パーティションの開始場所とLUNで使用するZFSブロック・サイズの最適な値を得る方法のガイドラインを示します。

FC LUNでのSun ZFSSAの使用

次の概要では、Sun ZFS Storage Applianceで利用可能なファイバ・チャネル・アーキテクチャ、アーキテクチャが機能する仕組み、およびシステム上でFC LUNを利用するための全体的な設定プロセスについて説明します。

Sun ZFSSAファイバ・チャネル・アーキテクチャ

Sun ZFS Storage Appliance用にサポートされているファイバ・チャネル・ホスト・バス・アダプタ（HBA）にはすべて、ポートが2個あります。したがって、各ユニットには、ファイバ・チャネルSANインフラストラクチャへの接続が2つあります。デュアル・ファブリック・ベースのSAN設計により、ホストへの完全なパスの冗長性が確保されます。

本書で説明する構成では、次の図に示すデュアル・ファブリックSANインフラストラクチャを利用する、Sun ZFSSAデュアル・ノード・クラスタを使用します。

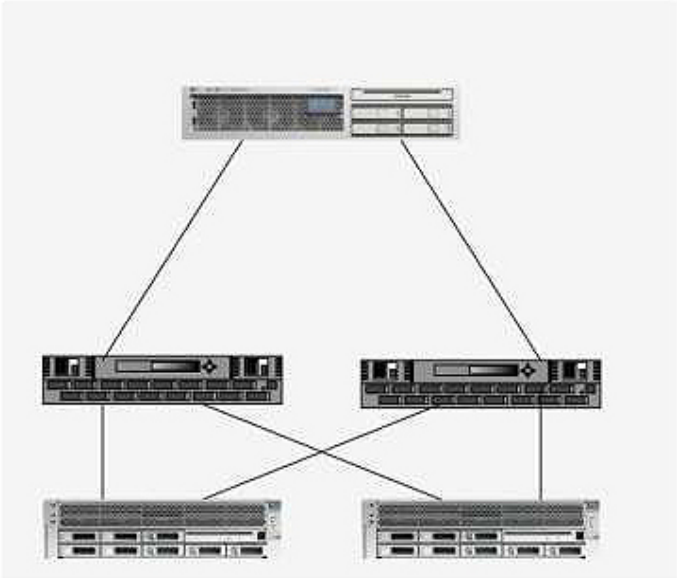

図1　SAN接続図

SANコンポーネント/パスのシングル・ポイント障害をなくすため、両方のホストHBAポート接続で常に、両方のSun ZFSSAノードを'監視'します。Sun ZFSSA上の各FC LUNについて、各ホストHBAパスは、アクティブ接続とスタンバイ接続の両方を監視します。

運用の理論

各Sun ZFSSAノードでは、"カスタマイズされた"Oracle Solarisオペレーティング・システムのインスタンスが稼働しています。障害の発生したI/Oパスから残っているアクティブ・パスへのI/Oリダイレクトを処理するマルチパス要素は、Common Multi-protocol SCSI Target（COMSTAR）フレームワークの一部です。

ターゲットやイニシエータ・グループ情報など、COMSTARフレームワーク内のすべての構成情報は、ノードのクラスタ機能によって、2つのSun ZFSSAノード間で常に同期されます。

Sun ZFSSAのプールで構成されたLUNは、Sun ZFSSAクラスタの両方のノードのFCポートに表示されます。プールを持つノードへの2つのパスは、アクティブ・パスとしてホストに表示され、プールを持たないノードを介したLUNへのパスには、スタンバイ・ステータスが表示されます。

したがって、Sun ZFSSAノードへのデュアルFC接続のように、ホストはLUNへのアクティブ・パス間でトラフィックのフェイルオーバーを開始できます。

アクティブ・パスとスタンバイ・パス間のフェイルオーバーは、Sun ZFSSAノード・フェイルオーバーによって開始できます。ノード・フェイルオーバーの前に、すべてのプールが、スタンバイ・パス・ステータスのあるノードにフェイルオーバーする必要があります。

ファイバ・チャネルの設定と構成の手順

ファイバ・チャネルのインストールと構成プロセスをできるだけ簡単に行うには、次のステップを記載された順番のとおりに実行してください。これらのステップの詳細は、次のセクションで説明します。

- Sun ZFSSAのハードウェアの準備
    1. 各ノードでSG-PCIE2FC-QF4（4Gb）またはSG-XPCIE2FC-QF8-Z（8Gb）デュアル・ポート搭載のFC HBAカードを、FC HBAの推奨スロットの場所に追加します。
    2. ノードの電源を入れて、HBAが認識されることを確認します。
    3. Sun ZFSSA HBAをターゲット・モードに設定して、FCポートのWorld Wide Name（WWN）を特定します。これらの名前は、SANスイッチでゾーンを設定するときに必要になります。
- ファイバ・チャネルのホスト接続
    1. ホストのFC HBAが、ホスト上のオペレーティング・システムによって認識されることを確認します。このホワイト・ペーパーの場合、Oracle Solarisホストは、1つのデュアル・ポートFC HBAで使用されます。
    2. ホストHBA FCポートのWWNを特定します。
- SANの構成とゾーニングの設定
    1. スイッチ、ホスト、Sun ZFSSAクラスタ間のケーブル配線を適切に行います。接続ライトがすべて点灯していることを確認します。接続ライトの点灯は、スイッチとのリンクが所望の速度で確立されたことを示します。
    2. HBAポートとSun ZFSSA FC HBAポートのWWNを使用して、ゾーニングを構成します。

3. FCターゲットおよびSun ZFSSAノード上のイニシエータ・グループを構成します。各ホストHBAポートのWWNを使用して、Sun ZFSSAノードへのアクセスを定義するには、ターゲット・グループとイニシエータ・グループをSun ZFSSAノードで構成する必要があります。

- Sun ZFSSAノード上でのLUNの作成と構成

1. LUNを作成して、ターゲット・グループとイニシエータ・グループのLUNへのメンバーシップを設定します。
2. Sun ZFSSAノード上のブロック・サイズやキャッシュ動作の要件といったLUNの属性が、要件どおりに設定されていることを確認します。属性はいつでも変更できます。ただし、LUNを一度作成したら、そのブロック・サイズ属性は変更できません。ブロック・サイズの値は十分に検討してから設定してください。
3. 構成済みのLUNがホスト上に表示されていることを確認します。
4. LUNのデバイス名のGUID（グローバル一意識別子）を、SUN ZFSSA BUIで表示されたGUIDと照合して検出します。
5. fdisk、format、またはpartedなど、ホストOSのツールを使用して、LUNをパーティション化して、ラベル付けします。
6. パーティションの開始場所と使用するファイル・システム間の適切なブロック・アライメントを、RAW LUNデバイスの開始場所に対して相対的に行います。詳しくは、「パーティション・アライメント」のセクションを参照してください。
7. SANゾーニング構成を確認します。

このホワイト・ペーパーでは、構成済みのLUNへのアクセスが確立されたことを確認するために、Oracle Solarisホストで使用できるコマンドについて説明します。Sun ZFSSAの非対称論理ユニット・アクセス（ALUA）機能が認識されるようにVMwareを設定する方法は、付録Bの「VMWare ALUAサポートの設定」を参照してください。

また、Sun ZFSSAオンライン・ヘルプ・マニュアルの「Configuring FC Client Multipathing」のセクションも参照してください。

注：このホワイト・ペーパーで後述するコマンドを実行するための適切な権限を確実に得るためには、Oracle Solaris上でroot権限を使用します。

ファイバ・チャネルLUNを利用するための設定プロセス

次に、FC LUNの設定と構成の詳細を説明します。

Sun ZFSSAハードウェアの準備

まず、FC HBAカードを各Sun ZFSSAノードに追加します。クラスタ構成で作業を行う場合、最初にすべてのサービスを残りのノードにフェイルオーバーして、FCカードを追加するノードの電源を切ります。

FC HBA、SG-XPCIE2FC-QF8-Z（8Gb）が、『Sun ZFS Storage 7x20 Appliance Customer Service Manual』に記載されているとおりに適切なスロットに配置されていることを確認します。

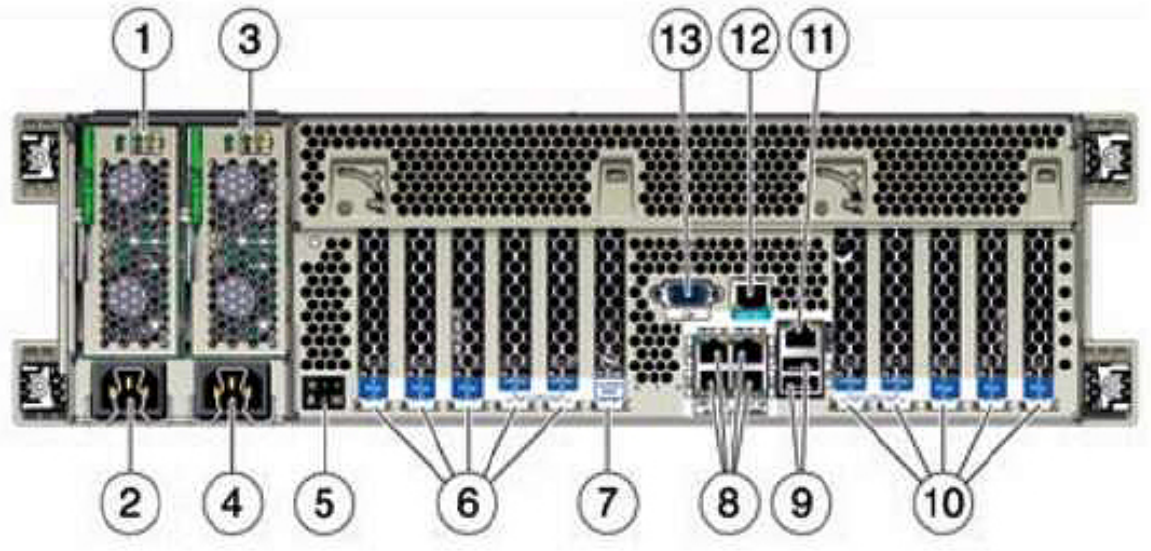

図2. 番号6と10のSun ZFSSA PCIeスロット

PCIスロットの番号は、Sun ZFSSAユニットの背面に表示されています（項目6と項目10）。

システムの起動後、GUIのハードウェア・ビューを使用して、FCカードが認識され、計画どおりにPCIeスロットに装着されているかどうかを確認します。

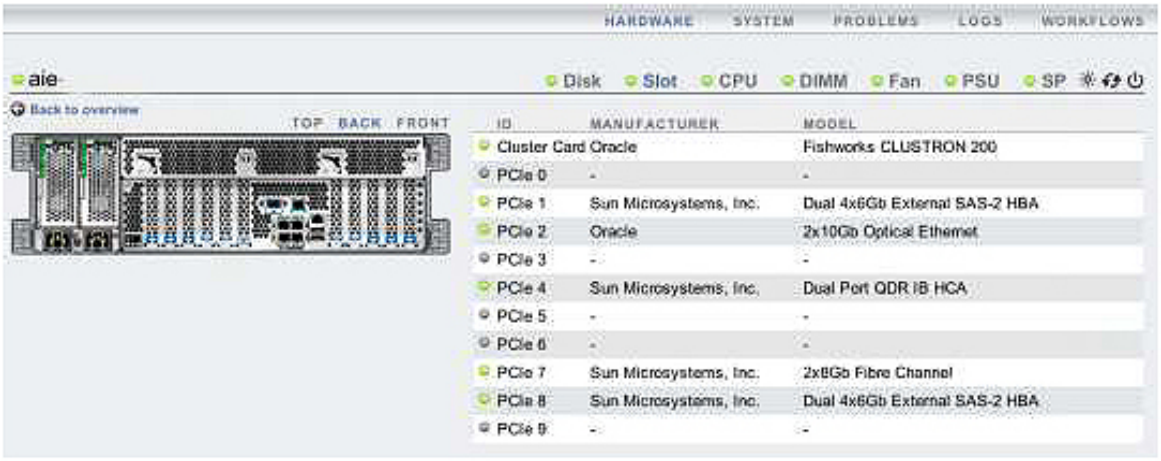

図3. Sun ZFSSA BUIでのHBAの確認

HBAがファブリックからアクティブな接続を受けているかどうかを確認するには、Configuration<SAN BUI ウィンドウに示されているように、ポート・アイコンの点灯している（赤）点を確認します。

図4. ポート・アイコンのアクティブ・リンク

ファイバ・チャネルのホスト接続

次に、ホストのオペレーティング・システム（この場合はOracle Solaris）での、構成済みのPCIeカード（FC HBA）の表示を確認します。HBA FCポートをSANインフラストラクチャに接続したら、ホストFC HBAの適切な機能を確立して、HBAポートのWWNを特定します。Oracle Solarisの場合、コマンド`cfgadm`を使用します。

`cfgadm -al`コマンドを使用して、HBA FCポートが利用可能であることを確認します。次のコード例は、HBAの出力を示します。

bash-3.2# cfgadm -al

```
Ap_Id  Type         Receptacle   Occupant     Condition
PCI0   scsi/hp      connected    configured   ok
PCI1   etherne/hp   connected    configured   ok
c14    fc-fabric    connected    unconfigured unknown
c15    fc-fabric    connected    unconfigured unknown
```

注：上のコード例では、関係のない情報の行は削除されています。

HBAファイバ・チャネル・ポートのWWNを特定するには、次のように、fcinfoコマンドを使用します。

```
bash-3.2# fcinfo hba-port
HBA Port WWN:210000e08b85b57b
     Port Mode:Initiator
     Port ID:10800
     OS Device Name:/dev/cfg/c14
     Manufacturer:QLogic Corp.
     Model:371-4522-02
     Firmware Version:05.04.03
     FCode/BIOS Version:BIOS:2.10; fcode:3.06; EFI:2.04;
     Serial Number:0402L00-1047843144
     Driver Name:qlc
     Driver Version:20110321-3.05
     Type:N-port
     State:online
     Supported Speeds:2Gb 4Gb 8Gb
     Current Speed:8Gb
     Node WWN:20000024ff24952e
```

```
      Max NPIV Ports:254
      NPIV port list:
HBA Port WWN:210100e08b85b57b
      Port Mode:Initiator
      Port ID:10900
      OS Device Name:/dev/cfg/c15
      Manufacturer:QLogic Corp.
      Model:371-4522-02
      Firmware Version:05.04.03
      FCode/BIOS Version:BIOS:2.10; fcode:3.06; EFI:2.04;
      Serial Number:0402L00-1047843144
      Driver Name:qlc
      Driver Version:20110321-3.05
      Type:N-port
      State:online
      Supported Speeds:2Gb 4Gb 8Gb
      Current Speed:8Gb
      Node WWN:20000024ff24952f
      Max NPIV Ports:254
      NPIV port list
```

上の出力から、ホスト・バス・アダプタのWWNとAP-ID（C14/C15）の両方を判別できます。AP-IDはデバイス・パスの識別子として使用されるため、Sun ZFSSAのイニシエータ・グループ設定で、イニシエータ名の一部として使用することをお勧めします。

SANの構成とゾーニングの設定

イニシエータとターゲットFC-HBAの両方を構成したので、SAN内の適切なルーティングをSANゾーンによって定義する必要があります。HBAが互いに表示されないようにするには、各ファブリックでイニシエータあたり1つ以上のゾーンが必要です。次の例では、論理接続あたり1つのゾーンが使用されます。

ゾーンを作成する前に、SAN構成が適切にケーブル配線されていることを確認してください。次のSAN構成では、最適な可用性が確保されます。色付きのラインは、論理'ゾーン化"接続を示します。

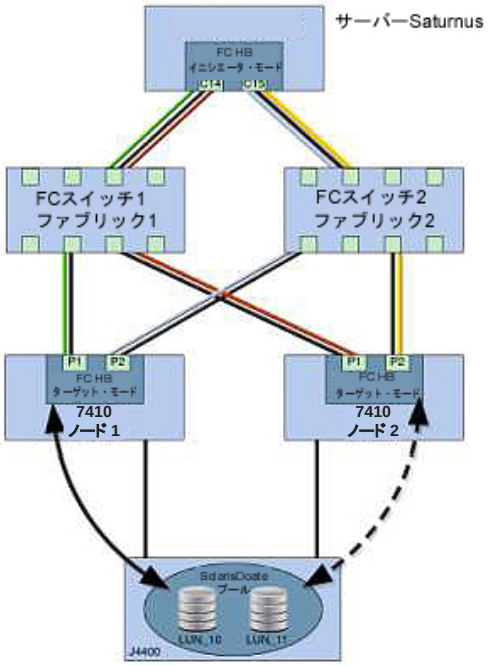

図5. SAN配線図

ストレージ・コントローラ（Sun ZFSSAノード）ごとにパスを必ず2つ設けてください。この冗長性により、スイッチとストレージ・コントロール間でSAN接続に障害が発生しても（自動的なノード・フェイルオーバーが開始されない状況）、停止が回避されます。

ゾーニングの方法には、WWNベースのゾーニングやSANスイッチ・ポート・ベースのゾーニングなど、さまざまな方法があります。

どれが望ましい方法かは、企業のITセキュリティ・ポリシーと構成管理ルールによって決まることがほとんどです。

例では、ポートWWNゾーニング方法が使用されます。エイリアス・オプションを使用して、スイッチのWWNに論理名を付けます。この例では、ホスト名とAP-IDの組合せをホストWWNとSun ZFSSAノード名のエイリアスに使用して、ポート名をSun ZFSSA WWNのエイリアスに使用します。

次の例では、4つのゾーンが使用されています。

表1. SANゾーニングの設定

ゾーン名	メンバー1		メンバー2	
ZONE A	Host1-HBA-C14	21:00:00:e0:8b:85:b5:7b	ZFSSA-NODE-1 Port 1	21:00:00:24:ff:31:82:66
ZONE B	Host1-HBA-C14	21:00:00:e0:8b:85:b5:7b	ZFSSA-NODE-2 Port 1	21:00:00:24:ff:31:83:04
ZONE C	Host1-HBA-C15	21:10:00:e0:8b:85:b5:7b	ZFSSA-NODE-1 Port 2	21:00:00:24:ff:31:82:67
ZONE D	Host1-HBA-C15	21:01:00:e0:8b:85:b5:7b	ZFSSA-NODE-2 Port 2	21:00:00:24:ff:31:83:05

このゾーン設定では、ホストとSun ZFSSAクラスタの間に、4つの論理パスを作成し、各ノードに2つずつパスを配します。ゾーンを適切に設定したら、Sun ZFSSA上でFCターゲットとイニシエータ・グループを構成できます。

Sun ZFSSAのインタフェース内でターゲット・グループの構成を使用して、Sun ZFSSAのターゲットFC HBAの特定のFCポートにLUNを割り当てます。ターゲット・グループは、ターゲットFCポートのプールとして機能します。外部からは、これらのポートでLUNを認識できます。

Sun ZFSSAのイニシエータ・グループは、LUNへのホスト（イニシエータ）アクセスを制御する手段として使用されます。したがって、イニシエータ・グループは、Sun ZFSSAのLUNにアクセスできるイニシエータFCポートのプールとして機能します。

図6のプルダウン・メニューの選択肢にあるように、まず、Sun ZFSSAで使用するファイバ・チャネルをTargetモードに設定します。

Sun ZFSSAノード内にアクティブなフェイルオーバー・パスを配し、もう一方のノードへノード・サービス・フェイルオーバー・パスを配するには、Sun ZFSSAクラスタの4つすべてのターゲット・ポートを、次のスクリーンショットで示すように、同じターゲット・グループ内で構成する必要があります。ノードを1つ構成してから、もう一方のノードに移って、そのノードの2つのターゲットFCポートを、最初のノード上ですでに作成済みのターゲット・グループに追加します。このステップは、画面上のドラッグ・アンド・ドロップ操作で実行できます。

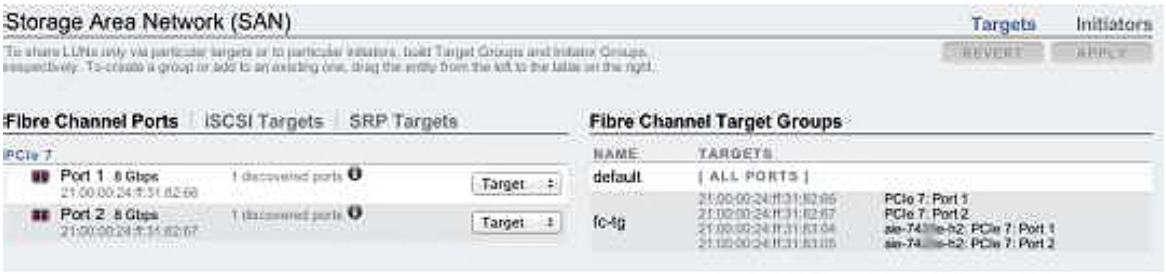

図6. FCターゲット・グループの設定

次に、ホスト・ポートをイニシエータ・グループに割り当てます。イニシエータ・グループは、設定プロセスの後半、LUNの作成時に使用します。イニシエータ・ポートに論理的で分かりやすい名前を付けます。

図7. FCイニシエータの設定

Sun ZFSSAノード上でのLUNの作成と構成

LUNを選択します。新しいLUNを追加するには、+記号を使用します。

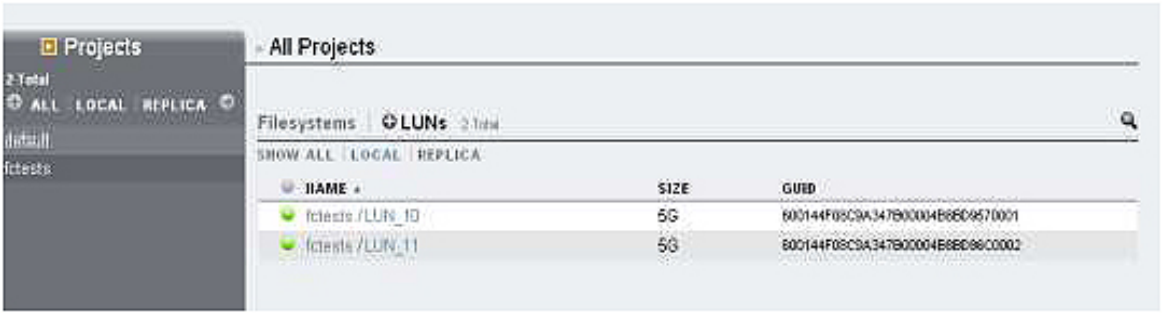

図8. ノードAへのLUNの追加

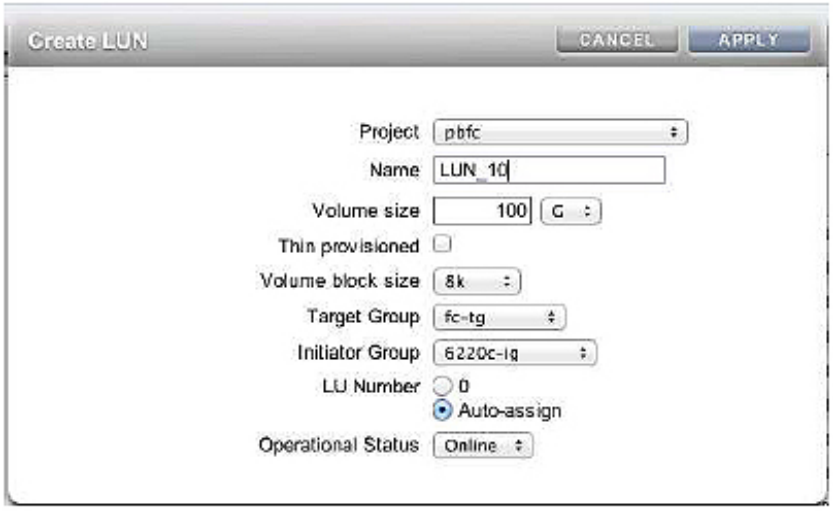

図9. FC LUNの仕様の設定

作成しているLUNのボリューム・サイズを指定します。ブロック・サイズの選択は慎重に行ってください。設定は、LUNを使用するアプリケーションによって用いられるI/Oアクセス・パターンのタイプによって左右されます。OLTPタイプのアプリケーションの場合は、普通のデータベース・ブロック・サイズをお勧めします。複数のブロック・サイズを使用するデータベースの場合は、レコード・サイズ用に最小のデータベース・ブロック・サイズを使用するか、異なるブロック・サイズの複数のLUNを使用します。REDOログ（ストリーミング・ワークロード）には、128Kのレコード・サイズを使用します。より大きいブロック・サイズ（最大128K）は、ストリーミングI/Oタイプのアプリケーションに適しています。LUNの作成後、これらの設定は変更できません。

LUNを割り当てるターゲット・グループとイニシエータ・グループを選択します。

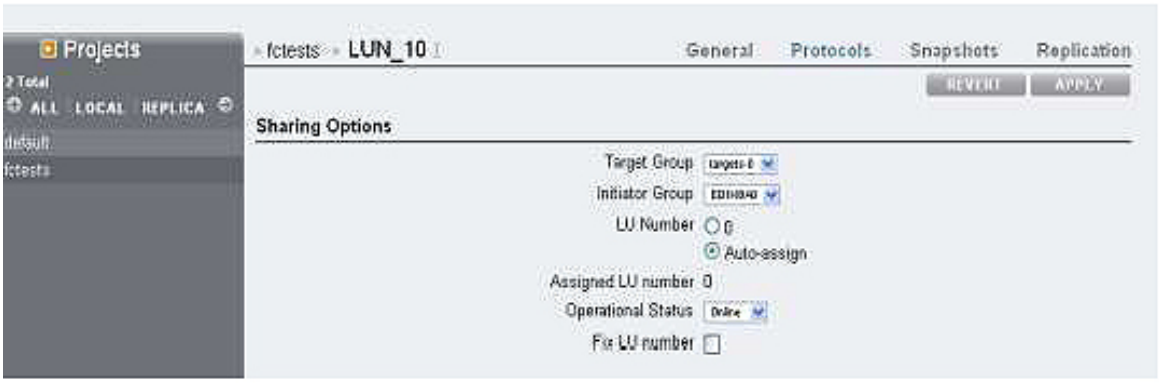

図10. LUNのアクセスの構成

各種LUNのプロパティは、Share (LUN)プロパティ・セクションのBUI Generalセクションで指定できます。LUNのパフォーマンス特性を微調整するため、表示されているこれらの各プロパティはいつでも変更できます。

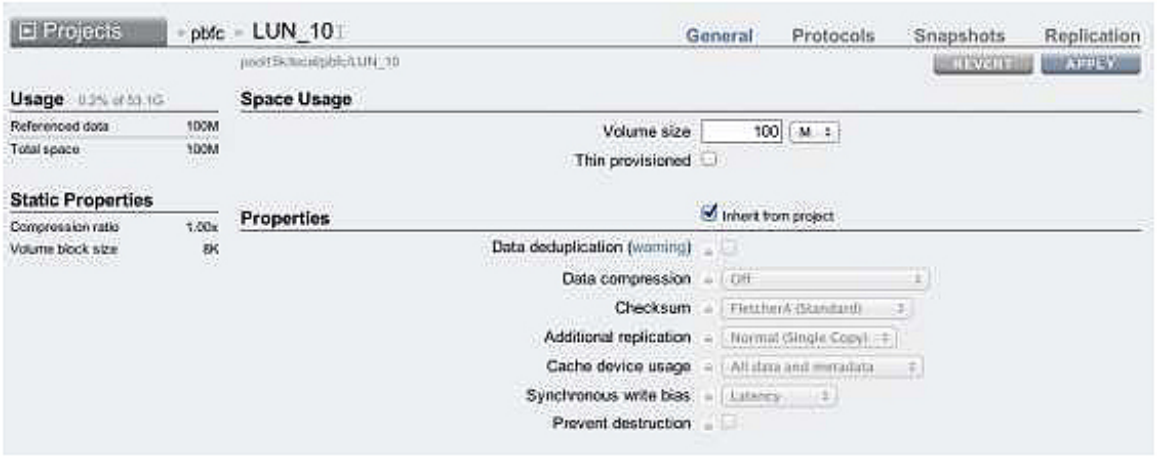

図11. LUNのプロパティの構成

ホスト上の構成の確認

`cfgadm -al`コマンドを使用して、HBAが構成されていることを確認します。構成されていない場合は、`cfgadm -c configure <Ap-ID>`コマンドを使用します。

次に、HBAのコマンド出力を示します。

```
bash-3.2# cfgadm -al
Ap_Id                  Type           Receptacle    Occupant      Condition
PCI0                   scsi/hp        connected     configured    ok
PCI1                   ethernet/hp    connected     configured    ok
c14                    fc-fabric      connected     configured    unknown
c14::21000024ff318266  disk           connected     configured    unknown
c14::21000024ff318304  disk           connected     configured    unknown
c15                    fc-fabric      connected     configured    unknown
c15::21000024ff318267  disk           connected     configured    unknown
c15::21000024ff318305  disk           connected     configured    unknown
```

上の出力では、ゾーニングが適切に設定されていることが示されています。C14は、両方のSun ZFSSAノードのポート1にアクセスでき、C15は両方のノードのポート2にアクセスできます。完全なパスの確認には、このセクションの後半に示すように、luxadmコマンドを使用します。

LUNのパーティション化とラベル付け

構成済みのLUNがホストで表示されることを確認するには、`format`コマンドを使用します。

```
bash-3.2# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
     0. c16t500000E010CC3B30d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
     /scsi_vhci/ssd@g500000e010cc3b30
     1. c16t500000E010CCD120d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
     /scsi_vhci/ssd@g500000e010ccd120
     2. c16t600144F0C0ACA00400004B8C13B60001d0 <SUN-SunStorage7410-1.0 cyl 160
     alt 2 hd 254 sec 254>
     /scsi_vhci/ssd@g600144f0c0aca00400004b8c13b60001
     3. c16t600144F0C0ACA00400004B8C13CC0002d0 <SUN-SunStorage7410-1.0 cyl 160
     alt 2 hd 254 sec 254>
     /scsi_vhci/ssd@g600144f0c0aca00400004b8c13cc0002
     4. c16t600144F08C9A347B00004B8BD96C0002d0 <SUN-SunStorage7410-1.0 cyl 160
     alt 2 hd 254 sec 254>
     /scsi_vhci/ssd@g600144f08c9a347b00004b8bd96c0002
     5. c16t600144F08C9A347B00004B8BD9570001d0 <SUN-SunStorage7410-1.0 cyl 160
     alt 2 hd 254 sec 254>
     /scsi_vhci/ssd@g600144f08c9a347b00004b8bd9570001
Specify disk (enter its number):
```

LUNを使用するには、ラベルを付ける必要があります。labelコマンドにより、LUNにパーティション表が作成されます。使用するパーティションのタイプおよびブロック・アライメントに関する考慮事項について詳しくは、「設計のベスト・プラクティス」のトピック「パーティション・アライメント」を参照してください。

SANゾーニング構成の確認

次に、SANのゾーニングとケーブル配線が適切に行われているかを確認します。LUNに対して4つのパスが表示されている必要があります。2つは、アクティブにLUNを処理しているSun ZFSSAノードへのアクティブなパス、2つは、もう一方のSun ZFSSAノードへのスタンバイのパスです。

利用可能なパスを見つけるには、`luxadm probe`コマンドを使用します。

```
bash-3.2# luxadm probe
No Network Array enclosures found in /dev/es
Found Fibre Channel device(s):
     Node WWN:500000e010cc3b30 Device Type:Disk device
       Logical Path:/dev/rdsk/c16t500000E010CC3B30d0s2
     Node WWN:500000e010ccd120 Device Type:Disk device
       Logical Path:/dev/rdsk/c16t500000E010CCD120d0s2
     Node WWN:2000001b32135c63 Device Type:Disk device
       Logical Path:/dev/rdsk/c16t600144F0C0ACA00400004B8C13B60001d0s2
     Node WWN:2000001b32135c63 Device Type:Disk device
       Logical Path:/dev/rdsk/c16t600144F0C0ACA00400004B8C13CC0002d0s2
     Node WWN:20000024ff318266 Device Type:Disk device
       Logical Path:/dev/rdsk/c16t600144F08C9A347B00004B8BD96C0002d0s2
     Node WWN:20000024ff318266 Device Type:Disk device
       Logical Path:/dev/rdsk/c16t600144F08C9A347B00004B8BD9570001d0s2
```

WWN `20000024ff318266`（Sun ZFSSAノードのHBAのWWN）からのデバイスが、対象のLUNです。

次の`luxadm display`コマンド出力で表示されるのは、1つのLUNの情報のみです。2つ目のLUNの情報も同一のはずです。出力では、LUN（Device Address 20000024ff318266,1）を処理するSun ZFSSAノードのポートに、LUNがアクティブに接続されていることが示されています。他方のノードへの2つの接続はスタンバイ状態になっています。

```
bash-3.2# luxadm display 20000024ff318266
DEVICE PROPERTIES for disk:2001001b322be2b4
     Vendor:                    SUN
     Product ID:                Sun Storage 74X0
     Revision:                  1.0
     Serial Num:
     Unformatted capacity:      5120.000 Mbytes
     Read Cache:                Enabled
       Minimum prefetch:        0x0
       Maximum prefetch:        0x0
     Device Type:               Disk device
     Path(s):
     /dev/rdsk/c16t600144F08C9A347B00004B8BD96C0002d0s2

     /devices/scsi_vhci/ssd@g600144f08c9a347b00004b8bd96c0002:c,raw


Controller                          /devices/pci@9,600000/SUNW,qlc@2,1/fp@0,0
   Device Address                   21000024ff318266,1
   Host controller port WWN         210100e08ba5b57b
   Class                            primary
   State                            ONLINE
```

```
Controller                      /devices/pci@9,600000/SUNW,qlc@2,1/fp@0,0
   Device Address               21000024ff318305,1
   Host controller port WWN     210100e08ba5b57b
   Class                        secondary
   State                        STANDBY

Controller                      /devices/pci@9,600000/SUNW,qlc@2/fp@0,0
   Device Address               21000024ff318267,1
   Host controller port WWN     210000e08b85b57b
   Class                        primary
   State                        ONLINE

Controller                      /devices/pci@9,600000/SUNW,qlc@2/fp@0,0
   Device Address               21000024ff318304,1
   Host controller port WWN     210000e08b85b57b
   Class                        secondary
   State                        STANDBY
```

書込みキャッシュに関する考慮事項

Sun ZFS Storage Applianceでは、ZFSファイル・システムが提供する機能に基づいて、大容量のデータ・キャッシュを使用します。

ZFSでは、メイン・メモリのAdaptive Replacement Cache（ARC）、およびソリッド・ステート・ドライブ（SSD）を使用した第2レベルのキャッシュからなる、2層のキャッシュ・システムが使用されます。この第2レベルのキャッシュは、Second Level Adjustment Replacement Cache（L2ARC）およびZFS Intent log（ZIL）という、読取りコンポーネントと書込みコンポーネントに分かれます。

図 12 ZFS キャッシュ・モデル

ZFSプールでのSSDの使用はオプションです。SSDを使用しない場合、ZILのブロックはZFSプールのディスク上で保持され、L2ARCの機能は利用できません。

L2ARC読取りキャッシュの機能は、ARCにブロックが検出されない場合に、ARCとディスク間の読取り待機時間を軽減することです。L2ARCはARCに比べて大きいため、保存または使用準備が整うまでしばらく時間がかかります。ランダム読取りワークロードには、L2ARCがもっとも適しています。

ZFS ZIL操作は常に、データ管理装置（DMU）トランザクションの一部になっています。DMUトランザクションが開くと、関連するZILトランザクションも開きます。これらのトランザクションは、安定したストレージにコミットされて、fsyncまたはO_DSYNC書込みが行われるまでメモリ内に蓄積されます。関連するZILトランザクションはほとんどの場合、DMUトランザクションがコミットされるときに破棄されます。

注：ZFSキャッシュ・メカニズムのより詳細な情報は、付録C「参考資料」で紹介するZFS開発者のさまざまなブログに掲載されています。

Sun ZFSSA上でLUNを作成する場合、パフォーマンスを向上させるために、アプライアンスのメイン・メモリを使用するか、データの一貫性を確実に維持するために、すべての書込みを安定ストレージに保存するかを選択できます。

書込みパフォーマンスが重要な場合は、ログ・デバイス（書込み最適化SSD）を構成します。データの一貫性を維持するには、LUNで書込みキャッシュを有効にしないでください。

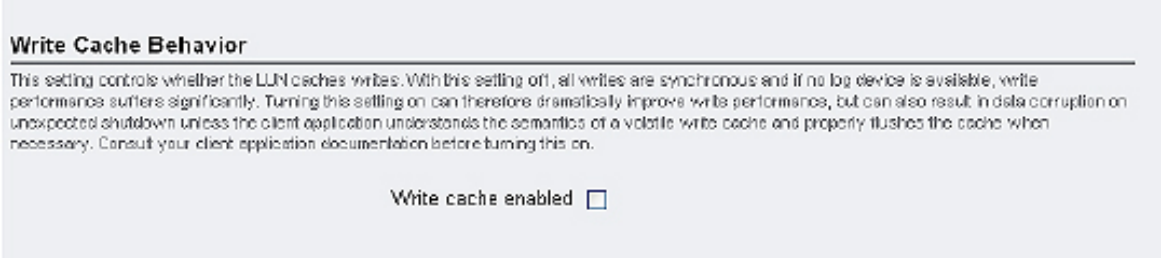

図13. 書込みキャッシュの動作

ZILの動作には、LUNごとに、synchronize write cache biasプロパティをlatency optimizedまたはthroughput optimizedに設定することで影響を与えることができます。これらの設定により、どのLUNが、自身が属するプールのログ・デバイス（書込み最適化SSD）を使用するかを制御できます。

Latency optimized：この設定は、書込みを書込み最適化SSDに即座にコミットして、待機時間を軽減します。書込みをSSDに即座にコミットすることで、書込みが障害から保護されます。この設定を使用すると、書込み最適化SSDの使用量が非常に多くなるため、デバイスのパフォーマンスによって制限されます。書込み最適化SSDをより多くストライプすると、秒あたりのI/O操作（IOPs）と帯域幅のパフォーマンスが改善されます。

Throughput optimized：書込みは書込みSSDアクセラレータをバイパスして、ディスクに直接コミットされます。この設定を使用すると、1つのトランザクションのトランザクション待機時間が増え、高度にマルチスレッド化された書込みがLUNに対し行われている場合に限り、適切なスループットを得られます。並列処理の負荷がこのように高くなるワークロードは少ないため、良好なパフォーマンス・レベルを維持するには、この設定を避ける必要があります。

ただし、このログ・バイアス設定を使用すると、データベース・ライターによってアクセスされるOracleデータ・ファイルを保存できます。この設定により、書込みバイアスSSDは、待機時間の少ないREDOログなど、他のより重要なI/Oを処理できるようになります。シングル・スレッドが大半の管理タスクの間、大きな速度の低下を避けるため、設定を切り替えることをベスト・プラクティスとしてお勧めします。

障害およびリカバリ・シナリオの構成のテスト

数多くのリンク障害への反応をテストするため、単純な負荷をホスト上に設定して、次の分析のスクリーンショットに示すように、1つのLUNから1つの読取りストリームを作ります。

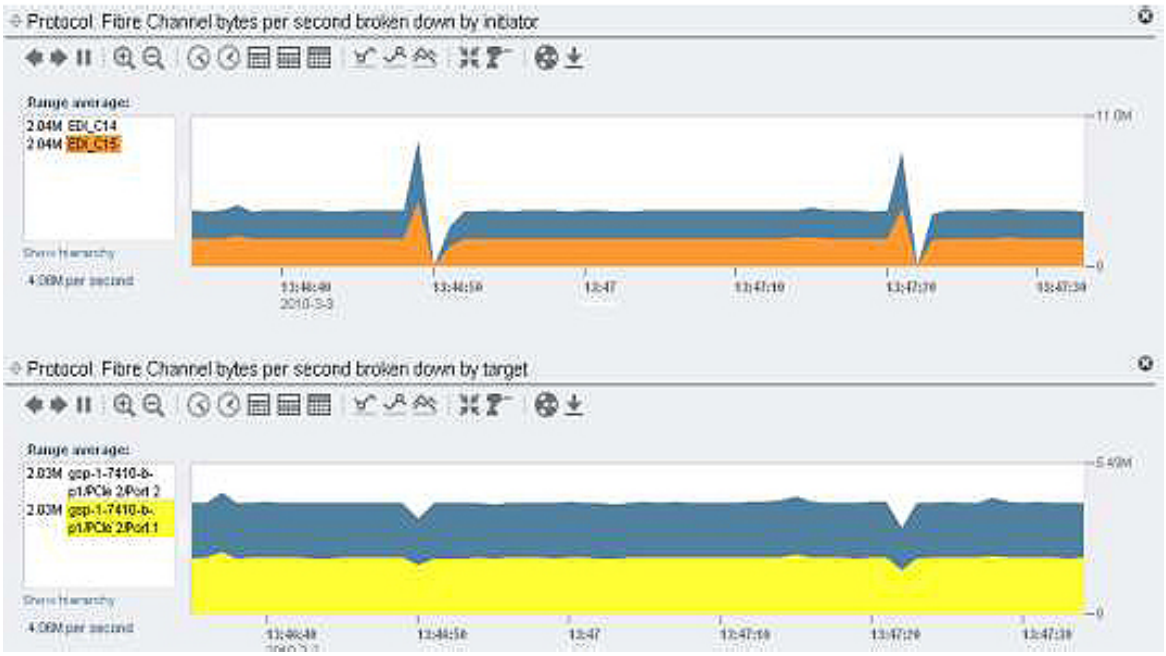

図14. 通常の状況下の負荷

この出力では、C14とC15に負荷が均等に分散され、I/O要求が1つのSun ZFSSAノードbのポート1とポート2に届いています。

ファブリックとホストHBAポート間のシングル接続障害

次のステップでは、C14からファブリックへの接続に障害を発生させます。次のスクリーンショットでは、トラフィックが停止せずに、残りの接続を介してFC LUNに継続的に流れている状態を示しています。リンクをリストアすると、LUNへのデュアル・パス・アクセスが正常に戻ります。

図15. 1つのリンクに障害が発生した状態

ファブリックとSun ZFSSAノード間のシングル接続障害

次のステップでは、Sun ZFSSAノードのアクティブ・ポートとファブリックの間にリンク障害を発生させます（16:13）。今回は、データの転送に小さな中断が生じ、残りのパスからのデータ転送は16:13:21に再開します。

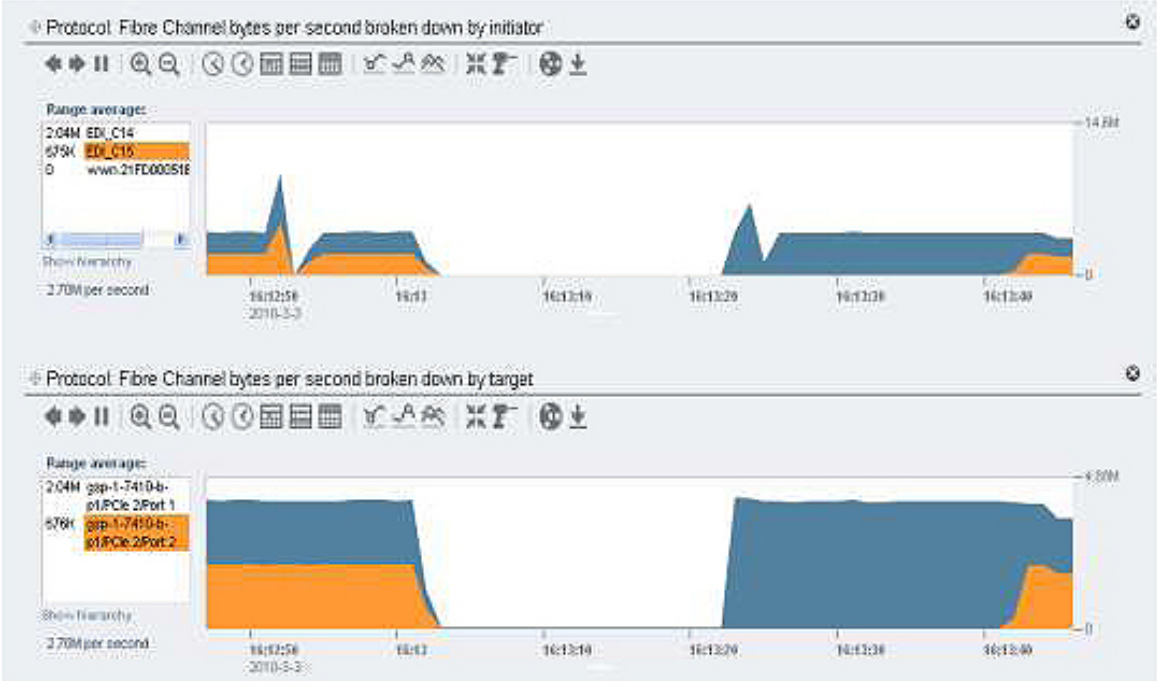

図16. アクティブなSun ZFSSAからのリンク切断

リンクが再開すると（16:13:40）、データ・フローはSun ZFSSAノードの両方のアクティブ・ポートに直接再分散されます。

Sun ZFSSAノードの両方のアクティブ・ポートへのリンク障害

これは、二重障害のシナリオです。FC LUNへのパスがあるSun ZFSSAへの両方のリンクに障害が発生すると、リンクが再確立されるまで、LUNへのI/Oは停止したままとなります。スタンバイ・パスを持つノードへのデータ・トラフィックのフェイルオーバーは、ノード・フェイルオーバーが開始されたときに実行されます。

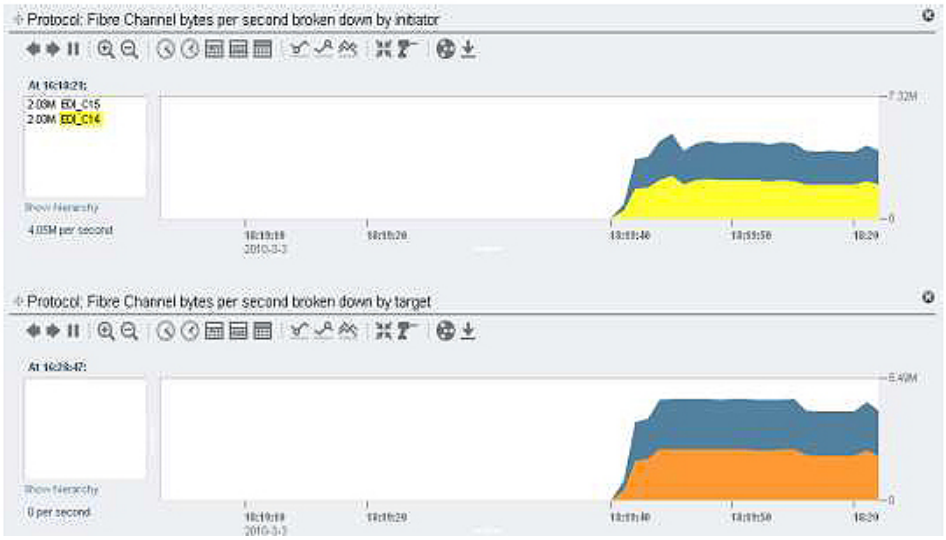

図17. Sun ZFSSAノードのフェイルオーバー

Sun ZFSSAノードの障害

FC LUNをアクティブに処理しているノードからのノード・テイクオーバーをトリガーすると、そのLUNのI/Oは、要求しているノードによって引き継がれます。1つのLUNのフェイルオーバーには、約30秒かかります。

障害シナリオのサマリー

テストの種類	結果
1つのホストからファブリックへのリンク障害	残りのホストHBAポートとSun ZFSSAノード上のアクティブ・ポート間のアクティブ・パスは引き続き、中断することなくI/Oを処理します。
Sun ZFSSAノードのアクティブ・ポートのリンクの1つに障害が発生	I/Oは約25秒間、一時的に停止してから、残りのアクティブ・パスで再開されます。パスがリストアされると、I/Oは瞬時に拾われます。
二重障害シナリオ。Sun ZFSSAノード上のアクティブ・ポートへの両方のリンクに障害が発生	I/Oは停止します。スタンバイ・パスを持つノードでトラフィックを再開するには、クラスタ・ノードのフェイルオーバーを開始する必要があります。
ユーザーが開始するノード・サービスのテイクオーバー	I/Oは約30秒間、一時的に停止します。その後、I/Oはもう一方のSun ZFSSAノードによって引き継がれます。

設計のベスト・プラクティス

ホスト接続に関する考慮事項

FC SANアーキテクチャにおける設計の重要な側面は、ストレージ・サブシステムに接続するホストの数、およびホストがSANのストレージ側のポートやLUNを共有するかどうかという点です。

ホスト間のLUNの共有は通常、ホスト・クラスタ・タイプの構成、あるいはLustreファイル・システムやOracle ASMストレージなどの、高性能計算（HPC）環境で行われます。複数のホスト間でLUNをアクティブに共有する場合、これらのホストのソフトウェアは、データ・アクセス競合の可能性を解決する役割を担い、適切に処理されない場合、データが破損する場合があります。ほとんどの場合、共有機能を持つファイル・システムを使用する必要があります。オラクルのQFSファイル・システムは、このような環境で機能するファイル・システムの一例です。ただし、これらのタイプのソリューションに関する説明は、このホワイト・ペーパーの対象範囲外です。

複数のホストが同じストレージ・サブシステムのポートに接続するアーキテクチャを扱う場合は、各ホストのSCSIキュー・メカニズムのサイズを慎重に考慮する必要があります。サイズを適切に決めることで、ストレージ・サブシステムのSCSIキューが、さまざまなホストの複数のSCSIコマンドで過負荷になるのを防ぎます。

コマンド・キュー深度の設定

ホストはコマンド・タグ・キューイング・メカニズムを通じて、一度に複数のI/OコマンドをLUNに実行できます。キュー内のコマンド数は、ベンダーのオペレーティング・システムと特定のFC HBAのドライバ実装によって異なります。

Sun ZFSSA FCターゲット・ドライバは、1つのHBAのキューで最大2048個のコマンドを処理できます。この特性から、アーキテクチャにある各ホストHBAポートのLUNあたりのキューの最大深度を導き出す必要があります。SCSIのタイムアウトによってパフォーマンスに悪影響が及ばないように、どのような場合でも、ターゲットHBAポートのキューのオーバーランを防ぐ必要があります。

LUNとホストFC接続が1つずつのシンプルな構成の場合、ホスト上の最大キュー深度は、2048を超えて設定することはできません。このLUNがn個のホストによって共有される場合、LUNを共有するホスト数でホストごとのキュー深度を割る必要があります。

Sun ZFSSA上で複数のLUNを構成する場合、ホスト上のLUNあたりのキュー深度は、2048に設定して、LUNの数で割る必要があります。これらのLUNが複数のホストで共有される場合、その数をさらにホスト数で割る必要があります。2048/n（LUN）×n（ホスト）で計算してから、もっとも小さい方の整数に丸めます。

最後に、Sun ZFSSAをアクティブ/アクティブ・クラスタ構成で使用するかどうかについて考慮します。ノードの1つが故障すると、残りのノードの対応するFCポートが、一方の側で構成されたLUNを処理します。キュー深度の計算では、安全のため、両方のクラスタ・ノードの対応するHBA側のLUNすべてを用います。

Sun ZFSSAクラスタに対する方程式は、2048/（n× *1*）または2048/（n× *1* ×2）となります。この場合、 *1* はSun ZFSSAターゲット・ポート上のLUNの数、nはLUNを共有するホストの数です。

Sun ZFSSAターゲット・ポートごとにこの計算を行ってください。

繰り返しますが、ターゲット・ポートに接続するLUNとホストの数を慎重に計画して、コマンド・キューの

過負荷を避けてください。複数のホストがLUNを共有する珍しいケースの場合、さらに多くの削減が必要になります。負荷に関わるフェイルオーバー構成内のすべてのポートについて考慮してください。

値は適当に設定することはできません。変更を行ったホストごとに再起動する必要があります。高い値ではなく、より低い値を選択して、控え目な値を使用してください。新しいLUNは後で追加できますが、すべてのホスト上のキュー深度の値を再び変更する必要があります。各ベンダーのオペレーティング・システムには、ホストの最大キュー深度を設定する上で、異なる構成メカニズムがあります。

OpenSolarisでのコマンド・キュー深度の設定

OpenSolarisの場合、2つのグローバル変数、`sd_max_throttle`と`ssd_max_throttle`によってキュー深度が設定されます。特定のHBAに対してどちらを使用するかは、HBAのドライバ自体がsdまたはssdドライバのどちらにバインドされているかによって決まります。いずれの場合も、変数は、ターゲットLUNごとに使用されるキュー深度を制御します。`s(s)d_max_throttle`のデフォルト設定は256です。したがって、Sun ZFSSA HBAポートごとに9個以上のLUNを使用する場合、`s(s)d_max_throttle`の値を低く設定する必要があります。

Solaris Kernelで`s(s)d_max_throttle`を設定するためのグローバル・メソッド

`s(s)d_max_throttle`を設定するには、次の行をカーネル・ファイル、/etc/systemに追加します。

```
set ssd:ssd_max_throttle=x
```

または

```
set sd:sd_max_throttle=x
```

この場合、前に説明したルールに従って計算したように、xはLUNあたりの最大キュー深度になります。

新しく構成したキュー深度をカーネルに使用させるには、システムを再起動する必要があります。

詳しくは、Sun ZFS Storage Applianceシステム・インタフェースのヘルプ・ファイルを参照してください。
https://<IPアドレス>:215/wiki/index.php/Configuration:SAN:FC#Queue_Overruns

パーティション・アライメント

ZFSボリュームは固定ブロック・サイズを使用して、データをボリュームに保存します。ブロックのサイズは、ボリュームまたはLUNの作成時に指定できます。理想的には、クライアントからの各I/O要求で同じブロック・サイズを使用し、これらの各ブロックを、ZFSボリュームの対応ブロックと同じ点で開始させる（位置合わせする）ことをお勧めします。クライアントからの読取りが同じサイズだが、位置合わせされていない場合、ZFSはデータをクライアントに返すために、2回以上ブロックにアクセスする必要があります。さらに、位置合わせされている書込みの場合、パーティションにずれがあると、これらの書込みは、本来は不要な読取りを発行して、いわゆるRead-Modify-Writeペナルティが生じる結果になります。

このずれにより、ディスクI/Oのオーバーヘッドが発生し、クライアントから見てLUNのパフォーマンスが低下します。この非効率性は、ランダムI/O OLTPタイプの環境では特に大きな影響を及ぼします。

アライメントのずれの原因

クライアントがLUNにアクセスできるようにするには、パーティション表をLUNに書き込む必要があります。パーティション化の実装は、オペレーティング・システムによって異なります。パーティション・スキームは従来どおり、物理ディスクのジオメトリ・パラメータ、セクタ、トラック、シリンダ、およびヘッドに基づきます。ストレージ・サブシステムのLUNの場合、これらのパラメータは完全に仮想的であり、ストレージ・サブシステムで使用されるディスクとの相関関係はありません。

そのため、ここでは、LUNで作成されるパーティションの開始セクタを、ストレージ・サブシステムで使用されるブロック・サイズに合わせて位置合わせすることが重要です。セクタのサイズは512バイトです。したがって、パーティションの開始セクタは必ず、1つのLUNブロック内の512バイト・ブロック数の倍数である必要があります。

次の例では、Sun ZFSSA ZFSボリュームが8kに設定されているため、パーティションの開始セクタは16の倍数にします。

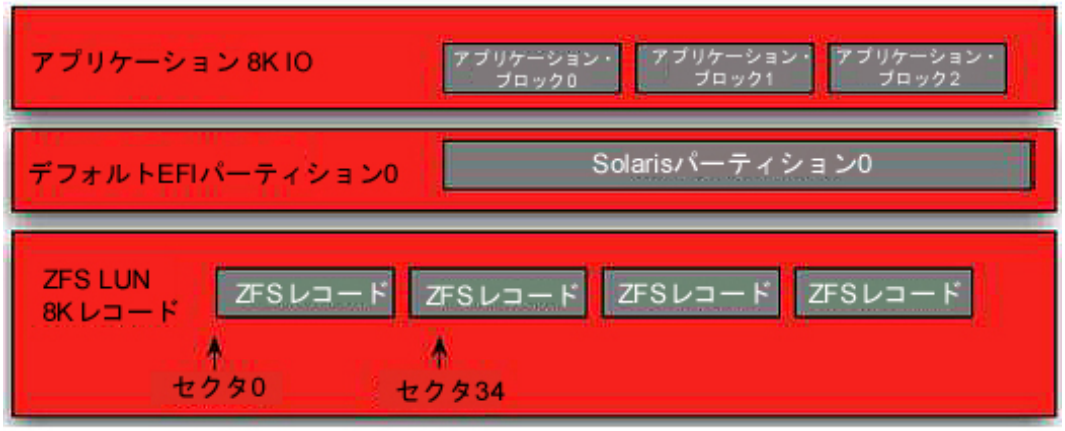

図18. Oracle ZFSSA FC LUN上のOracle SolarisのデフォルトのEFIパーティション・スキーマ

パーティション化ツールで作成された開始セクタが34だと、アライメントにずれが生じます。128kのZFSブロック・サイズの場合、開始セクタは256の倍数にする必要があります。

OSのタイプによって、使用されるパーティション化スキームは異なります。x86ベースのオペレーティング・システムでは、旧DOSのfdiskツールに基づくパーティション化スキームが使用されます。LinuxとVMwareでは依然として、このタイプのパーティション化スキームが使用されています。

業界のベスト・プラクティスでは、1MBのパーティション・オフセット・スキームを推奨する方向へと動いているので、2048の倍数であるパーティションの開始セクタを使用してください。ZFSベースのLUN/ボリュームの場合、最大ZFSブロック・サイズ（現時点では128k）と同サイズ以上のパーティション化オフセット（開始セクタ）を選択してください。

特定のLUN用にZFSで使用するブロック・サイズは、次のダイアログ・ウィンドウでLUNを作成するときに指定します。

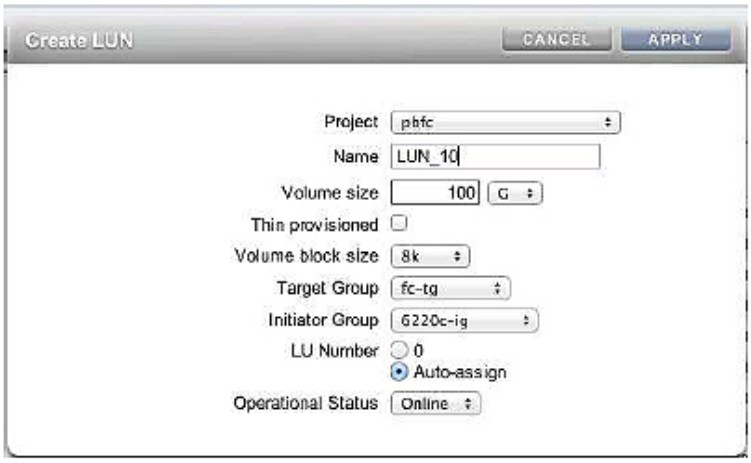

図 19. LUN用にZFSで使用するブロック・サイズの指定

Oracle Solarisのパーティション化

Oracle Solarisでは、Storage Management Initiative（SMI）標準とExtensible Firmware Interface（EFI）標準の2つのパーティション化スキームを使用します。SMIは、Oracle SPARC Solarisに由来し、シリンダのパーティションの開始と終了を表します。EFI標準では、セクタ・ベースのスキームを使用します。

次のコードは、EFIラベルを指定するパーティションの作成を示します。

```
format>label
[0] SMI Label
[1] EFI Label
Specify Label type[1]:1
Ready to label disk, continue? y
format> partition
       ~~~~~~~~~~~~~~
partition> print
Current partition table (original):
Total disk sectors available:209698782 + 16384 (reserved sectors)
Part      Tag             Flag         First Sector  Size          Last Sector
0         usr             wm           34            99.00GB       207618081
1         unassigned      wm           0             0             0
2         unassigned      wm           0             0             0
3         unassigned      wm           0             0             0
4         unassigned      wm           0             0             0
5         unassigned      wm           0             0             0
6         unassigned      wm           0             0             0
7         unassigned      wm           0             0             0
8         reserved        wm           209698783     8.00MB        209715166


partition> 0
Part    Tag               Flag         First Sector  Size          Last Sector
0       usr               wm           34            99.00GB       207618081
Enter partition id tag[usr]:
Enter partition permission flags[wm]:
Enter new starting Sector[34]:256
Enter partition size[207618048b, 207618303e, 101376mb, 99gb, 0tb]:
```

```
partition> pri
Current partition table (unnamed):
Total disk sectors available:209698782 + 16384 (reserved sectors)

Part   Tag                 Flag         First Sector  Size          Last Sector
0      usr                 wm           256           99.00GB       207618303
1      unassigned          wm           0             0             0
2      unassigned          wm           0             0             0
3      unassigned          wm           0             0             0
4      unassigned          wm           0             0             0
5      unassigned          wm           0             0             0
6      unassigned          wm           0             0             0
7      unassigned          wm           0             0             0
8      reserved            wm           209698783     8.00MB        209715166
partition>
```

複数のパーティションを使用する場合、EFIスキームはより細かい粒度を提供して、ブロック・アライメントを保証します。EFIベースの最初のパーティション開始はデフォルトで34です。ただし、この設定では、どのZFSブロック・サイズでもアライメントにずれが生じます。選択したどのZFSブロック・サイズでもアライメントを適切に維持するには、開始セクタ番号をZFSの最大ブロック・サイズ、128Kにします。この設定により、最初のパーティションの開始セクタが256になります。

x86プラットフォーム上のOpenSolarisでは、EFIベースのパーティション・スキームのみがサポートされます。fdiskベースのPCパーティション化スキームの上部で使用する必要があります。fdiskカービングはここではパーティションと呼ばれ、Sun EFIスキーマはスライスと呼ばれる点に注意してください。したがって、fdiskパーティション化スキーマ内には、1つまたは複数のスライスがあります。これらのスキーマは両方とも、各パーティションの開始とパーティション内のスライスの開始がZFSブロック・サイズに合わせて位置合わせされるように設定する必要があります。PCのパーティション化の詳細は、次の「Windowsのパーティション化」、「VMwareのパーティション化」、および「Linuxのパーティション化」のセクションを参照してください。

OpenSolaris環境では、Gnome Partition Editor（GParted）ツールの(g)partedを使用して、PCのパーティション化スキーマを作成および変更できます。

次の図は、FCから送られた読取りI/Oとディスクからの読取りI/Oのグラフを示します。

第1期間では、EFIラベルのLUNが、セクタ256で始まるパーティションで使用されています。第2期間では、EFIによって生成されたパーティションのデフォルトの開始セクタ34が使用されています。

第1期間の場合、ディスクからのI/Oの数とFC LUNのI/Oの数は、1対1の関係になっています（1850 IOPS）。

第2期間の場合、FC LUNの読取りごとに、2つのディスク読取りI/Oが生成され、その結果、FC I/OブロックをZFSブロック・サイズに合わせて位置合わせすると、FC LUN上のI/Oの数の方が少なくなります。

図20. EFIラベルのLUNの開始セクタ256と開始セクタ34の比較

Windows OSのパーティション化

パーティション・アライメントは、インストールされたWindowsオペレーティング・システムのバージョンによって異なります。

Windows Server 2008とWindows Vista

Windows Server 2008とWindows Vistaの場合、パーティション・アライメントはデフォルトで1MBに設定されます（4GB超のディスクの場合）。レジストリで使用されている値は、次の参照先で確認できます。

HKLM¥SYSTEM¥CurrentControlSet¥Services¥VDS¥Alignment

Windows Server 2003以前

Windows 2003 Server以前のWindows環境で作成されたパーティションは定義により、位置合わせは行われていません。デフォルトで使用されるパーティション表の開始は、32256バイトです。Windowsの動的なディスクの場合、dmdiag -vtoコマンドを使用して、パーティションのオフセットを確認します。基盤のディスクの場合は、 wmicコマンドを使用します。

作成済みのFC LUN上にパーティションを作成するには、diskpartツールを使用します。次のテンプレートを使用すると、パーティションのオフセットを設定し、ドライブ・レターを割り当て、ファイル割当て単位のサイズ（unit=<xxK>）を指定できます。

```
Diskpart
list disk
select disk <DiskNumber>
create partition primary align=<Offset_in_KB>
assign letter=<DriveLetter>
format fs=ntfs unit=64K label="<label>" nowait
```

推奨される128K以上、またはその倍数のオフセット値を使用してください。

ファイル割当て単位のサイズは、アプリケーションで推奨されるWindowsファイル・システムの使用方法によって異なります。同じ値のZFSブロック・サイズを使用してください。

VMwareのパーティション化

VMware環境の場合、仮想マシンはVirtual Machine File System（VMFS）を使用し、PC fdiskパーティション内で作成されます。作成したパーティションを、基盤のZFSSA ZFS構造に合わせて位置合わせすることが重要です。VMwareのESXハイパーバイザを使用してVMFSを作成する場合、デフォルト値はESXバージョンに応じてわずかに異なります。

ゲスト・オペレーティング・システムのデータ・ファイル・システムは、VMFS最上部の仮想マシン・ディスク形式（VMDK）で保存されます。ファイル・システムはVMFSに合わせて位置合わせする必要があります。ゲスト・オペレーティング・システムは、RAWデバイス・マッピングも使用できます。RAWデバイス・マッピング上のパーティション・アライメントは、LUNを直接使用するOSの場合と同じです。ゲストOSのブート・パーティション・アライメントは不要です。

アライメント・モデルを簡素化するには、VMFSを保持するLUN上のパーティション、ゲストOSのVMDKを保持するVMFS最上部のパーティションの両方で、256セクタ・オフセットを使用します。

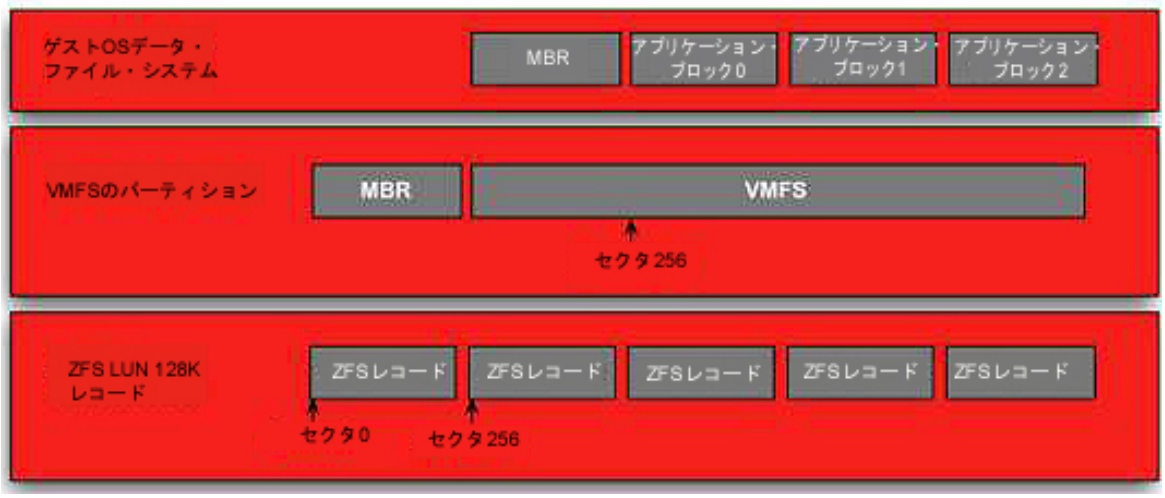

図21. VMware VMFSおよびVMDKのパーティション・アライメント

VMware ESX 3.0

エキスパート・モードでfdiskツールを使用して、パーティション1の必要な開始ブロックを指定します。

VMware ESX 4.0とESXi 4.0

vSphere Clientのvmkfstoolsプログラムは、64KB境界に沿ってパーティションを自動的に位置合わせします。ただし、どのZFSブロック・サイズも自動的に位置合わせされるわけではありません。128K境界の使用をお勧めします。

VMware VMFSブロック・サイズ

VMFSブロック・サイズのサイズによって、VMFSインスタンスの最大サイズが決まります。ゲストOS（ファイル・システム）が使用するI/OサイズとVMFSブロック・サイズの間に関連性はありません。ゲストOSの読取り操作と書込み操作は、vmkernelによって外部ストレージ・サブシステムに透過的に送られます。したがって、使用するZFSブロック・サイズの値は、ゲストOSのファイル・システム/アプリケーションが使用するI/Oサイズに応じて決める必要があります。多くのファイル・システムでは、4Kブロック・サイズが使用されます。

Linuxのパーティション化

Linuxのfdiskパーティション化ツールはシリンダも使用して、パーティションの開始と終了を定義します。シリンダの開始位置とサイズは、基盤のZFSファイル・システムのブロックに合わせて調整する必要があります。Linuxのfdiskツールの`-S`オプションにより、トラックあたりのセクタ数を指定できます。ZFSブロック・サイズの倍数である値を使用すると、アライメントが適切に調整されます。

ただし、fdiskの`-S`オプションでは、64以上の値は許可されません。ほとんどの場合、デフォルトで使用される値は63です。その結果、1KB超のZFSブロック・サイズでは、必ずアライメントがずれてしまいます。

実際的な解決策としては、32を指定して、`fdisk -u`コマンドを使用することで、セクタで表されるパーティションの開始値を指定します。最初のパーティションでは256を指定し、続くすべてのパーティションでは`+n`を使用して、256の倍数を使用します。ここで、`n`は、パーティションのセクタ数を表します。この解決策は、すべてのZFSブロック・サイズに対応します。

注：fdiskルーチンにより、パーティションがシリンダ境界で終了（または開始）しないことを告げる警告が表示されることがあります。シリンダは単にツールによる生成物にすぎないので、これらの警告は無視してかまいません。

```
root@petertest /root % fdisk -u -S 32 /dev/sda

The number of cylinders for this disk is set to 5140

Command (m for help):n
Command action
       e      extended
       p      primary partition (1-4)
p
Partition number (1-4):1
First sector (32-41942399, default 32):256
Using default value 256
Last sector, +sectors or +size{K,M,G} (256-41942399, default 41942399):
Using default value 419424399

Command (m for help):p

Disk /dev/sda:21.4 GB, 21474836480 bytes
255 heads, 32 sectors/track, 5140 cylinders, total 41942400 sectors
Units = sectors of1* 512 bytes
Disk identifier:0x000da626
Device      Boot        Start       End         Blocks      Id          System
/dev/sda1   *           256         41942399    106080      83          Linux
```

```
Command (m for help):w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
root@sysresccd /root % fdisk -l -u /dev/sda

Disk /dev/sda:21.4 GB, 21474836480 bytes
255 heads, 32 sectors/track, 5140 cylinders, total 41942400 sectors
Units = sectors of 1 * 512 = 512 bytes
Disk identifier:0x000da626

  Device    Boot        Start       End         Blocks      Id          System
/dev/sda1   *           256         41942399    2097120     83          Linux
root@petertest /root %
```

前述したように、業界のプラクティスでは、1MBサイズのパーティション・オフセットのスキーム・アライメントに移行します。したがって、2048の倍数であるパーティションの開始セクタを使用してください。

SSDキャッシュ・タイプのガイドライン

参照テストの設定

パフォーマンス・テストを実行すると、各種キャッシュ構成の動作をよりよく理解できます。I/Oワークロードは、アプリケーションのタイプ、およびアプリケーションで実行されるワークロードのタイプによって異なります。アプリケーションでは、日中はランダム書込みワークロードの割合が高く、夜間はある種のデータ・マイニング・アクティビティの形でバックアップが実行されると同時に、順次読取り指向のワークロードが増加するかもしれません。

さまざまなOLTPタイプのワークロードを提示するため、次の表に示すように、それぞれのワークロードの種類を使用して、I/O動作を測定します。完全なテスト・セットでは、8種類のワークロードを実行して、秒あたりのI/O結果を取得します。

I/Oのタイプ	データの局所性	順次	ランダム
100%読取り	100%キャッシュ	IOPS	IOPS
	非キャッシュ	IOPS	IOPS
100%書込み	100%キャッシュ	IOPS	IOPS
	非キャッシュ	IOPS	IOPS

データ・カテゴリの局所性については、Sun ZFSSAの適応型置換キャッシュ（ARC）におけるデータ・キャッシュを、100%または極めて低いヒット率のいずれかで測定しました。100%のキャッシュ・ヒット率を得るため、I/OはLUNの小さい領域（この場合は1GB）でI/Oを実行しました。LUNの領域全体でI/Oを実行するため、キャッシュがまったくヒットしない状態を作りました。LUNのサイズは100GBであり、Sun ZFSSAで使用されたサイズよりはるかに大きいキャッシュ・サイズです。

非同期と同期の書込み間の相違を測定するため、dsync open flagオプションあり/なしの状態で、書込みテストをLUNで実行しました。

パフォーマンス・テスト・ツールには、Vdbenchを使用しました。このツールは非常に柔軟性に優れ、細かく指定および制御されたワークロードに対応できます。使用したワークロードは、付録BのVDBENCHパラメータ・ファイルに示されています。

テスト中、ワークロードによってLUNへのキュー・オーバーランが起こらないように、最大32のスレッドを使用しました。比較点を追加するため、75%と25%の読取りの負荷を使用して、OLTPタイプのワークロードを実行しました。

テストは、次の構成におけるプール内のLUNで実行されています。

- SSDキャッシュ・デバイスなし

- 読取り最適化キャッシュ・デバイス（L2ARC）を使用
- 書込み最適化キャッシュ・デバイス（ZIL）を使用
- 読取りおよび書込み最適化キャッシュ・デバイスの組合せを使用

注：テスト・パフォーマンスの設定は、FCターゲット環境で使用されたSun ZFS Storage Applianceの最大パフォーマンスを示すものではありません。極端な設定のワークロードについてSun ZFSSA ARC、L2ARC、およびZILの動作を比較するために、数値を提供することのみが目的です。

テストではもう1つパラメータが取り入れられており、テスト中、LUNでSun ZFSSA書込みキャッシュをオンまたはオフにしました（最初のセクション「FC LUNでのSun ZFSSAの使用」の「書込みキャッシュに関する考慮事項」を参照）。「書込みキャッシュに関する考慮事項」のセクションの図13に示すように、このオプションにより、書込みキャッシュでのARCの使用がオンまたはオフになります。

注：LUN書込みキャッシュ・オプションは使用しないことを強くお勧めします。このオプションを使用すると、データ整合性に問題が生じる可能性があります。このオプションを有効にする場合は、このリスクを理解して、データを別のソースからリカバリできるようにしておく必要があります。

テストLUNは100GB、8KB ZFSブロック・サイズで、ミラー化されたRAID構成の完全なJ4400ラック（24台のデバイス）を使用したプールからのLUNです。ARC2テストでは、1台の読取り最適化キャッシュ・デバイスと2台の（ミラー化された）書込み最適化キャッシュ・デバイスを使用しています。

テスト結果の解釈

次のセクションでは、各種構成のパフォーマンス・テスト結果を検証して、さまざまなパフォーマンス目標を達成する上での最適な設定について結論を導きます。

Sun ZFSSAの書込みキャッシュの使用

次のグラフでは、SSDタイプのキャッシュ・デバイスを使用せず、LUN書込みキャッシュ・オプションの値を変化させた場合のLUNのテスト結果を比較します。

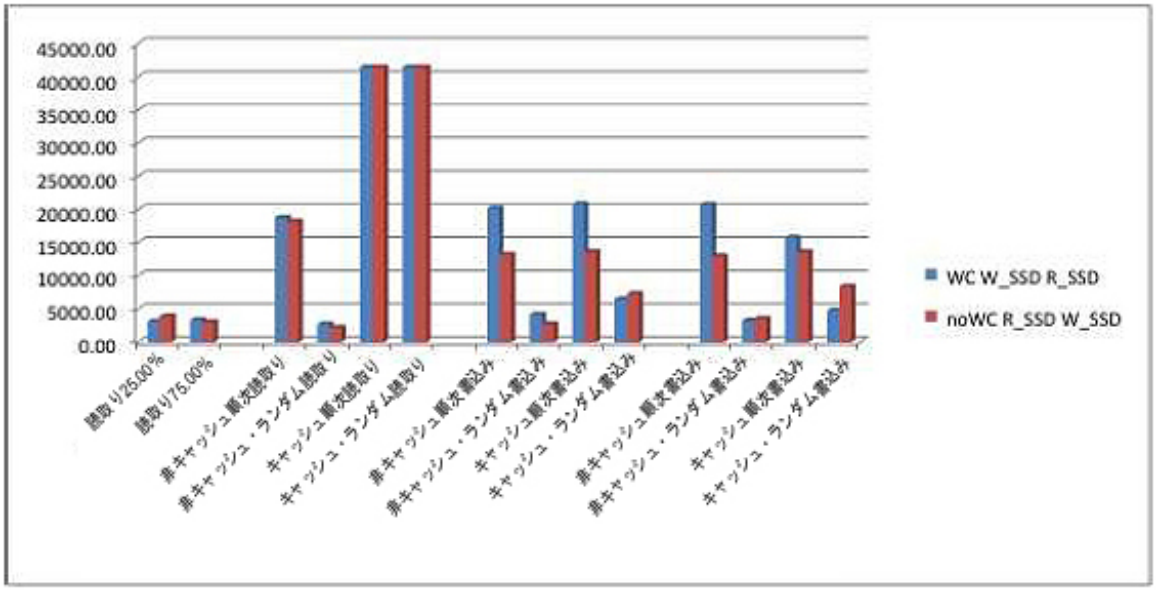

図22. LUNでのLUN書込みキャッシュ・オプションの比較（SSDデバイスなし）

予測したように、読取りパフォーマンスはLUN書込みキャッシュ設定の有効化または無効化による影響を受けません。書込み操作の場合、大きな違いがあります。

次のグラフでは、書込み最適化SSDデバイスを、LUNが構成されているプールに追加した場合のパフォーマンスの比較を示します。

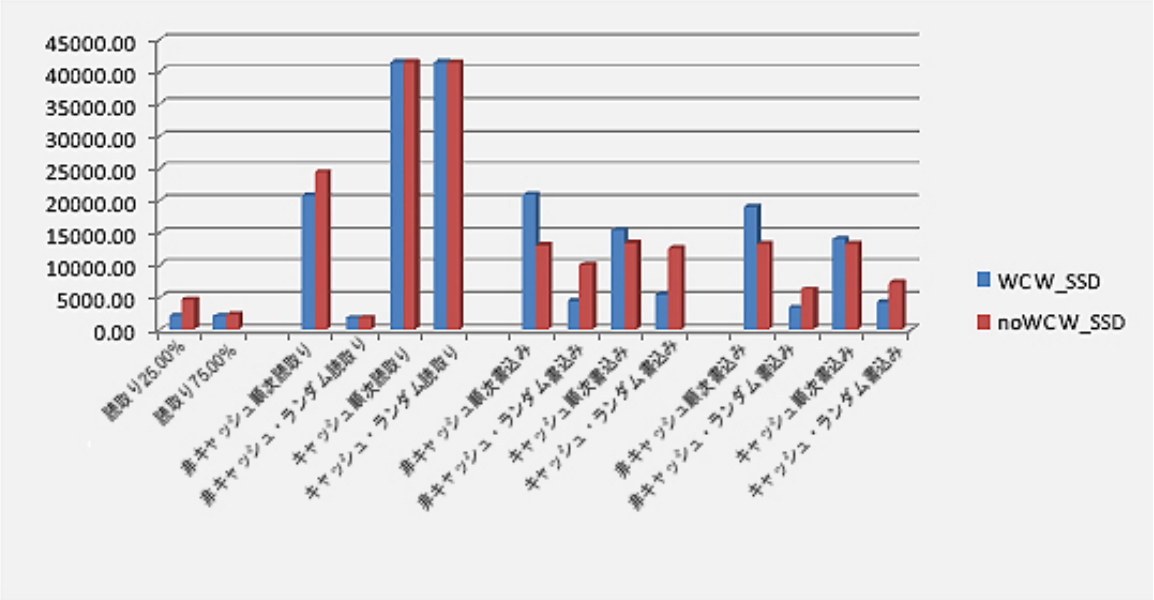

図23. LUNでのLUN書込みキャッシュ・オプションの比較（SSDデバイスを追加した場合）

書込み最適化SSDオプションを使用すると、ARCがなくても、書込みキャッシュを大幅に補完します。したがって、書込みパフォーマンスとデータ整合性が重要な場合は、書込み最適化SSDオプションを使用すると、LUN書込みキャッシュが有効に設定されている場合と同じレベルのパフォーマンスが得られます。さらに重要なことに、書込み最適化SSDオプションを使用すると、ローカル・メモリ・キャッシュ速度を使用するシステムのパフォーマンス・レベルで、システムのデータ整合性が維持されます。書込みキャッシュ・デバイスの数は、アプリケーション・ワークロードの予想される'局所性'に合わせて調整する必要があります。

SSDキャッシュ・デバイス・タイプの比較

各種SSDキャッシュ・タイプで実施されたテスト結果を比較すると、どのタイプのSSDキャッシュ・タイプ・デバイスをいつ使用すればいいのかをより的確に把握できます。

明らかな所見として、キャッシュされた読取りI/Oタイプのワークロードの場合、SSDタイプ・キャッシュ・デバイスが何の影響も及ぼさないように見える点が挙げられます。ただし、このテストで使用されたように、1つのLUN上の負荷の場合、このことは確実です。すべての読取りデータがSun ZFSSA ARCに収まります。複数のLUN（およびボリューム）が使用される場合、読取りタイプのSSDを使用して、Second-level Read Cache（L2ARC）を作成できます。

もう1つの非常に重要な結論は、ARCでのキャッシュ・ヒット率が高い、ランダム指向タイプの書込みワークロードを大量に扱う場合、'cache device usage'を'meta data only'または'none'に設定することをお勧めします。

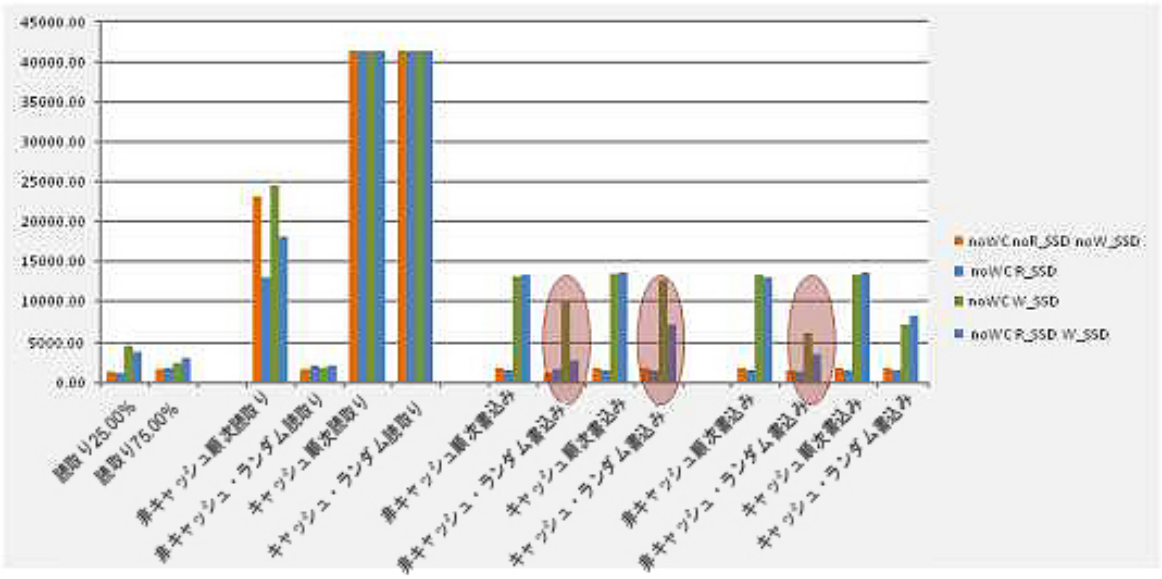

図24. さまざまなSSDキャッシュ・タイプ・オプションを使用したLUNのパフォーマンス

OLTPタイプのワークロード

次のグラフは、テストされた75%の読取りワークロードを前面（最初の列）に、25%の読取りワークロードを背面（2番目の列）に示します。さらに、読取り機能の以前の結論と所見も示します。

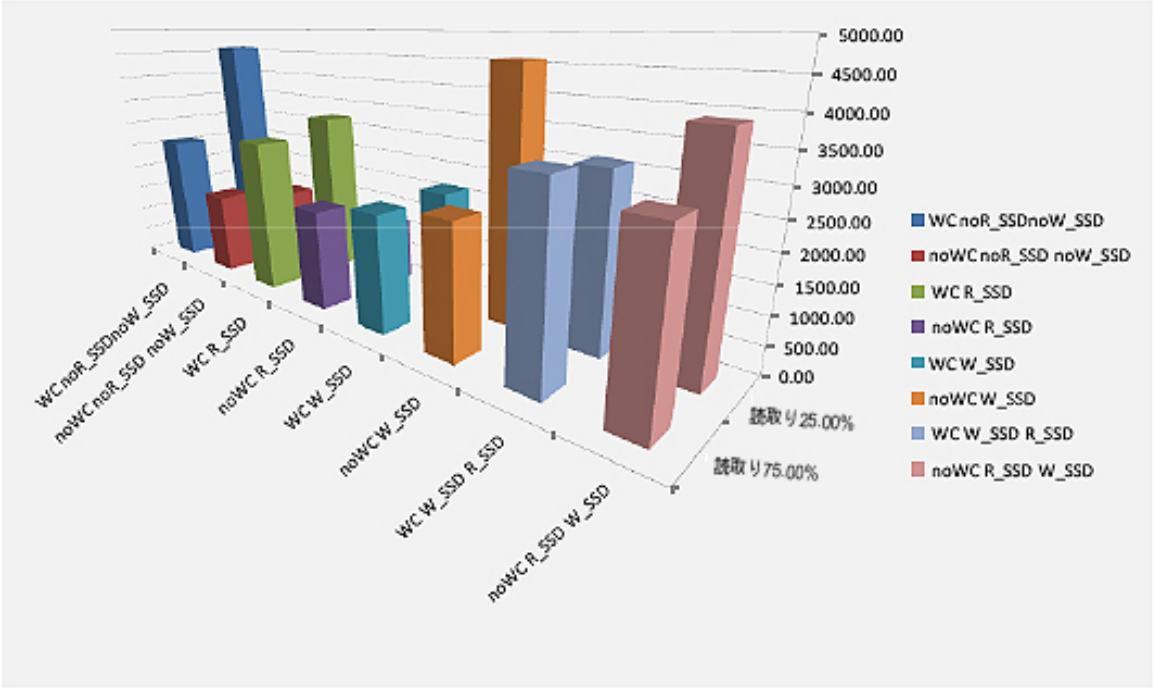

図25. OLTPタイプ・ワークロードでのSSDタイプ・キャッシュ・オプションのLUNパフォーマンスの比較

結論

結果は、SSDタイプ・キャッシュ・デバイスを構成するときに考慮する必要のある2つの主要な要因を示しています。

- ランダム書込み集中型I/Oロードの場合、‘cache device usage’を‘meta data’または‘none’に設定します（図27を参照）。
- .LUN書込みキャッシュ・オプションを使用しない場合、適切なタイプと数のSSDタイプ・キャッシュ・デバイスを構成することで、パフォーマンスの影響を補うことができます。

一般に、使用するLUN/ボリュームのI/Oパフォーマンス特性に合わせてプールを調整します。作成しているプールのRAIDプロファイル・オプションを選択するときには、Sun ZFS Storage ApplianceのBUIの推奨事項に従ってください。BUIには、利用可能なプロファイル・オプションのそれぞれについて簡単な説明が表示されます。

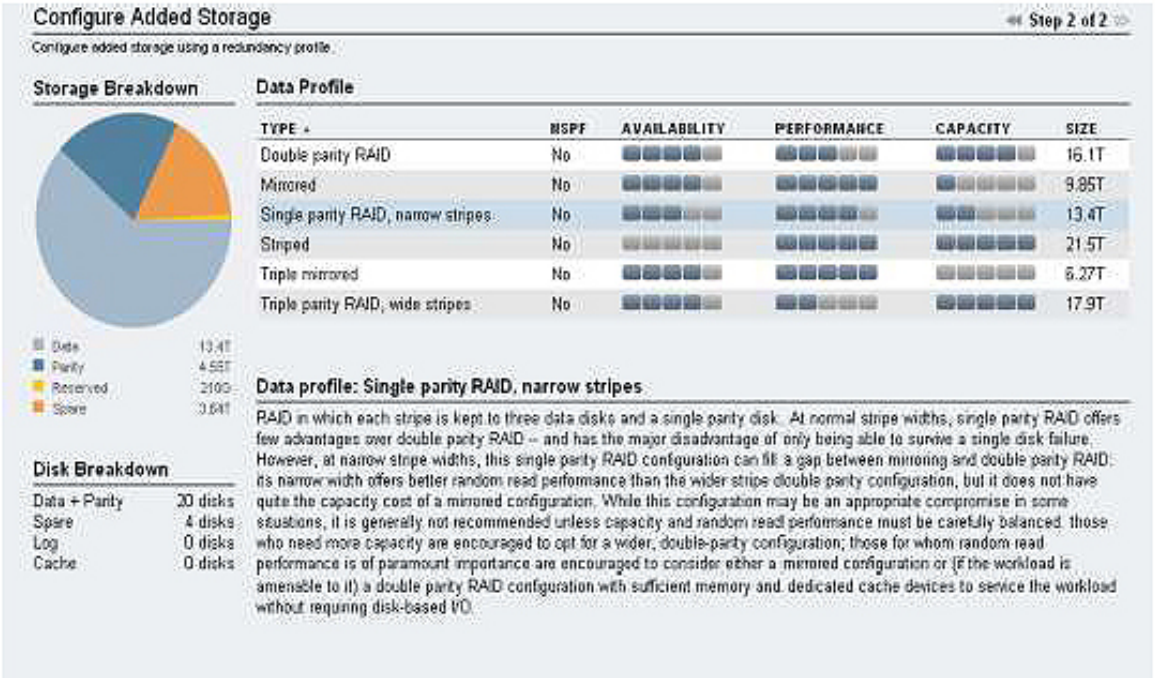

図26. BUI、データ・プロファイル・オプションおよび説明

LUN/ボリューム単位で、SSDタイプ・キャッシュ・デバイスの使用に影響を及ぼすことができます。共有または単一のLUN/ボリュームのいずれかのプロパティ・オプションの一部として、SSDキャッシュ・デバイスの読取りと書込みの動作を次の表に表示されているように指定できます。

プロパティ	オプション	結果
Cache device usage	All data and metadata	すべてのデータ、このプロジェクト/LUN/ボリュームで読取りタイプのSSDが使用される
	Metadata only	すべてのメタデータ、このプロジェクト/LUN/ボリュームで読取りタイプのSSDが使用される
	Do not use cache devices	このプロジェクト/LUN/ボリュームで読取りタイプのSSDは使用されない
Synchronous write bias	Latency	このプロジェクト/LUN/ボリュームで書込みタイプのSSDが有効になる
	Throughput	このプロジェクト/LUN/ボリュームで書込みタイプのSSDは使用されない

SSD関連のプロパティは、プロジェクトまたはLUN/ボリュームのいずれかのPropertiesウィンドウで指定できます。

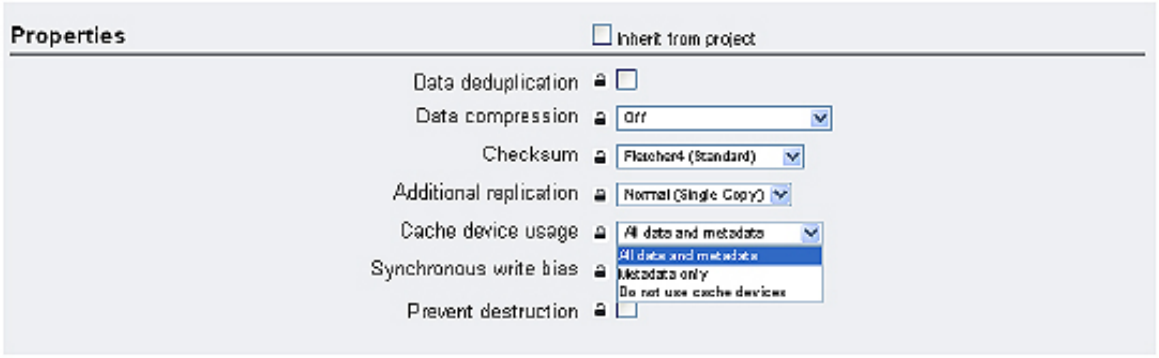

図27. 共有/ボリューム/LUNのプロパティでのSSDキャッシュ・タイプの動作の指定

付録A：vdbenchパラメータ・ファイル

vdbenchバージョン5.02のツールを使用して、このホワイト・ペーパーで使用するパフォーマンスの数値を生成しました。1つ留意すべきことは、要求されたブロックに書込みがまったく行われなかった場合、ZFSは、ディスクまたはキャッシュから読取りを行うことなく、クライアントに直接読取り要求を返す点です。そのため、読取りの数値は当然、非常に良好な値になりますが、あまり現実的な値ではありません。LUNをSun ZFSSAで作成するたびに、dd書込みストリームをそのLUNで実行してから、そのLUNでパフォーマンス・テストを開始しました。

次のvdbenchワークロード・スクリプトを使用しました。

```
# LUNs
sd=LUN1_8k,hitarea=1GB,threads=32,lun=/dev/rdsk/c16t600144F08C9A347B00004BACF5290008d0s2
sd=LUN1_8k_sync,hitarea=1GB,threads=32,openflags=o_dsync,lun=/dev/rdsk/c16t600144F08C9A3
47B00004BACF5290008d0s2



# Following measurements:
#
#          cached                non-cached
# read     sequential, random    Sequential, Random
# write    sequential, random    Sequential, Random
#
# Cached is done by using 1GB of the LUN for the measurement, as defined by hitarea in
sd definition, see above
#
# Only 8kB blocks are used.
# To simulate a DB IO, last test is run for 75% and 25% reads.
#  with O_DSYNC using 8kB blocks

wd=smallio,sd=(LUN1_8k)
wd=smallio_sync,sd=(LUN1_8k_sync)


#
rd=ReadsSmallIO,wd=smallio,warmup=200,rdpct=100,forrhpct=(0,100),forseekpct=(0,100),forx
fersize=(8k),forthreads=(32),iorate=max,elaps=90,interval=10,pause=10

rd=WritesSmallIO,wd=smallio,warmup=200,rdpct=0,forwhpct=(0,100),forseekpct=(0,100),forxf
ersize=(8k),forthreads=(32),iorate=max,elaps=90,interval=10,pause=10
rd=WritesSmallIO_sync,wd=smallio_sync,warmup=200,rdpct=0,forwhpct=(0,100),forseekpct=(0,
100),forxfersize=(8k),forthreads=(32),iorate=max,elaps=90,interval=10,pause=10


# DB mix load
rd=DBLoad,wd=smallio_sync,warmup=200,rdpct=75,forrhpct=(25,75),whpct=5,seekpct=100,xfers
ize=8k,forthreads=(32),iorate=max,elaps=90,interval=10,pause=10
```

付録B：VMware ALUAサポートの設定

VMware ESX 4.0のドライバは、Sun ZFSSA FCマルチパスALUA機能をデフォルトで認識しません。ESX 4.1のドライバはALUAに対応します。

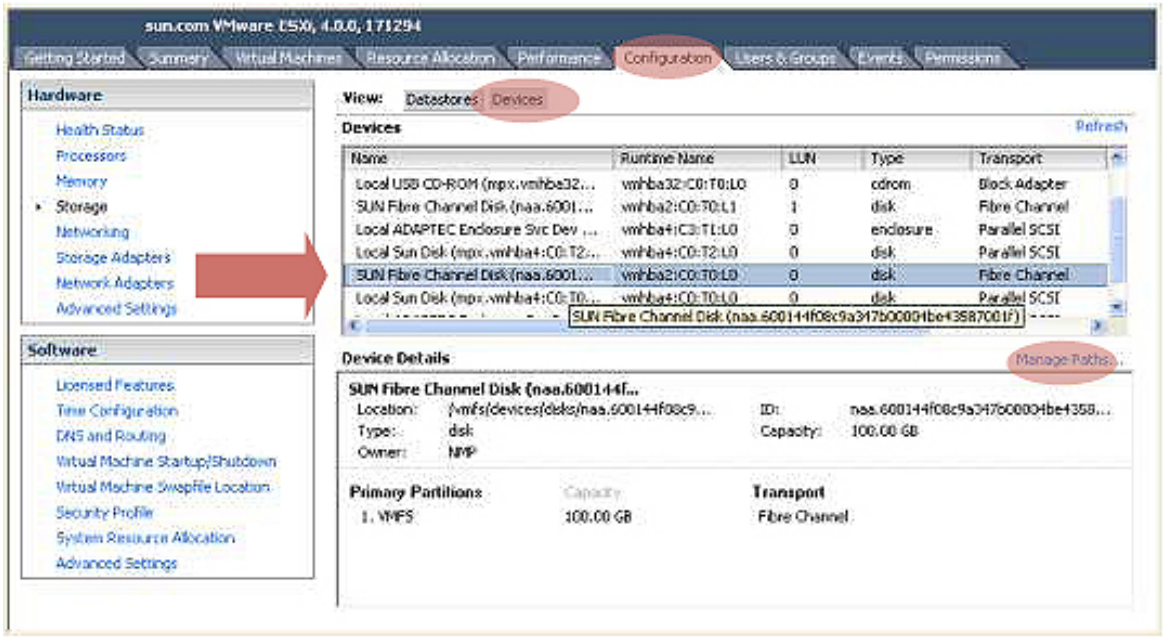

図 28

FC LUNが利用可能になると、スタンバイ・パスには、‘Dead’のステータスが表示されます。VMwareにSun ZFSSA FC LUNスタンバイ・パスを認識させるには、構成ルールをSATPプラグインに追加する必要があります。ファイバ・チャネルを使用して、Sun ZFSSAクラスタ構成に接続する各VMware ESX(i)サーバーに、このルールを追加します。

次に説明するコマンドがESX(i)サーバーのコマンドライン・インタフェースから実行されます。SSHを使用して、サーバーへのアクセス権を取得するか、コンソール上のCLIインタフェースを使用します。

現在のSATPプラグイン・ルールの確認

次のesxcliコマンドを実行します。

```
~ # esxcli nmp device list
naa.600144f08c9a347b00004be43587001f
       Device Display Name:SUN Fibre Channel Disk
       (naa.600144f08c9a347b00004be43587001f)
       Storage Array Type:VMW_SATP_DEFAULT_AA
       Storage Array Type Device Config:
       Path Selection Policy:VMW_PSP_FIXED
       Path Selection Policy Device Config:
{preferred=vmhba2:C0:T1:L0;current=vmhba2:C0:T0:L0}
       Working Paths:vmhba2:C0:T0:L0
```

コマンド出力でSUN Fibre Channel Diskエントリを見つけてください。VMW_SATP_DEFAULT_AAとしてStorage Array Typeが示されています。このアレイ・タイプの定義には、ALUA機能は含まれていません。

ベンダーおよびモデルのident情報の判別

ベンダーとモデルのident情報は、VMwareカーネルのメッセージ・ファイルに記載されています。

```
~ # fgrep SUN /var/adm/messages | fgrep Storage
May   7   18:00:06   vmkernel:1:00:51:19.169   cpu13:5070)ScsiScan:839:   Path
'vmhba2:C0:T0:L0':Vendor:'SUN ' Model:'Sun Storage 7410' Rev:'1.0 '
```

前のコマンドを使用すると、VMwareカーネルによって検出されたSun ZFSSA FC LUNの複数のテキスト行が返されます。Sun ZFSSAの各モデルで、モデル番号情報が異なることに留意してください。

SATP構成ルールの追加と確認

次のコマンドを実行することで、ベンダーとモデルに続くテキストを使用して、VMwareにSun ZFSSA ALUA機能を認識させます。

```
~ # esxcli nmp satp addrule -s VMW_SATP_ALUA -e "Sun Storage 7410" -V "SUN" -M
"Sun Storage 7410" -c "tpgs_on"
~#
```

この例では、Sun ZFSSA 7410モデルが使用されています。次のコマンドを実行して、ルールが正しく追加されたことを確認します。

```
~# esxcli nmp satp listrules | fgrep SUN | fgrep ALUA
VMW_SATP_ALUA SUN Sun Storage 7410 tpgs_on Sun Storage 7410
~#
```

Sun ZFSSA FC LUN ALUAパス・ステータスの確認

ESXサーバーを再起動します。または、FC LUNがまだ検出されていない場合は、FC HBAの再スキャンに進んで、新しいLUNを追加します。

サーバーの起動後、使用されているプラグインを再確認します。いったん有効にすると、新たに検出されたすべてのLUNがALUAプラグインによって取得されます。

```
~ # esxcli nmp device list
naa.600144f08c9a347b00004be43587001f
       Device Display Name:SUN Fibre Channel Disk
       (naa.600144f08c9a347b00004be43587001f)
       Storage Array Type:VMW_SATP_ALUA
       Storage Array Type Device Config:
{implicit_support=on;explicit_support=off;explicit_allow=on;alua_followover=on;{
TPG_id=0 ,TPG_state=AO}{TPG_id=1,TPG_state=STBY}}
       Path Selection Policy:VMW_PSP_MRU
       Path Selection Policy Device Config:Current Path=vmhba3:C0:T0:L0
       Working Paths:vmhba3:C0:T0:L0
```

パス・ステータスを確認します。この構成例では、1つのLUNに4つのパスがあります。2つはアクティブで、2つはスタンバイです。「SANの構成とゾーニングの設定」セクションで示されているものと同じ構成設定が使用されます。

```
~ # esxcli nmp path list

fc.2001001b322b5eb6:2101001b322b5eb6-fc.2001001b32335c63:2101001b32335c63-
naa.600144f08c9a347b00004be43587001f
       Runtime Name:vmhba3:C0:T1:L0
       Device:naa.600144f08c9a347b00004be43587001f
       Device Display Name:SUN Fibre Channel Disk
       (naa.600144f08c9a347b00004be43587001f)
       Group State:standby
       Storage Array Type Path
       Config:{TPG_id=1,TPG_state=STBY,RTP_id=256,RTP_health=UP} Path Selection
       Policy Path Config:{non-current path}

fc.2001001b322b5eb6:2101001b322b5eb6-fc.2001001b322be2b4:2101001b322be2b4-
naa.600144f08c9a347b00004be43587001f
       Runtime Name:vmhba3:C0:T0:L0
       Device:naa.600144f08c9a347b00004be43587001f
       Device Display Name:SUN Fibre Channel Disk
       (naa.600144f08c9a347b00004be43587001f)
       Group State:active
       Storage Array Type Path
       Config:{TPG_id=0,TPG_state=AO,RTP_id=2,RTP_health=UP}
       Path Selection Policy Path Config:{current path}

fc.2000001b320b5eb6:2100001b320b5eb6-fc.2000001b32135c63:2100001b32135c63-
naa.600144f08c9a347b00004be43587001f
       Runtime Name:vmhba2:C0:T1:L0
       Device:naa.600144f08c9a347b00004be43587001f
       Device Display Name:SUN Fibre Channel Disk
       (naa.600144f08c9a347b00004be43587001f)
       Group State:standby Storage Array Type Path
       Config:{TPG_id=1,TPG_state=STBY,RTP_id=257,RTP_health=UP}
       Path Selection Policy Path Config:{non-current path}

fc.2000001b320b5eb6:2100001b320b5eb6-fc.2000001b320be2b4:2100001b320be2b4-
naa.600144f08c9a347b00004be43587001f
       Runtime Name:vmhba2:C0:T0:L0
       Device:naa.600144f08c9a347b00004be43587001f
       Device Display Name:SUN Fibre Channel Disk
       (naa.600144f08c9a347b00004be43587001f)
       Group State:active
       Storage Array Type Path
       Config:{TPG_id=0,TPG_state=AO,RTP_id=1,RTP_health=UP}
       Path Selection Policy Path Config:{non-current path}
~#
```

vSphere Clientを再起動します。パス情報は‘Configuration’タブにあります。このタブに移動するには、デバイス・ビューを使用して、「Fibre Channel disk」を選択し、「Manage Path」オプションを使用します。

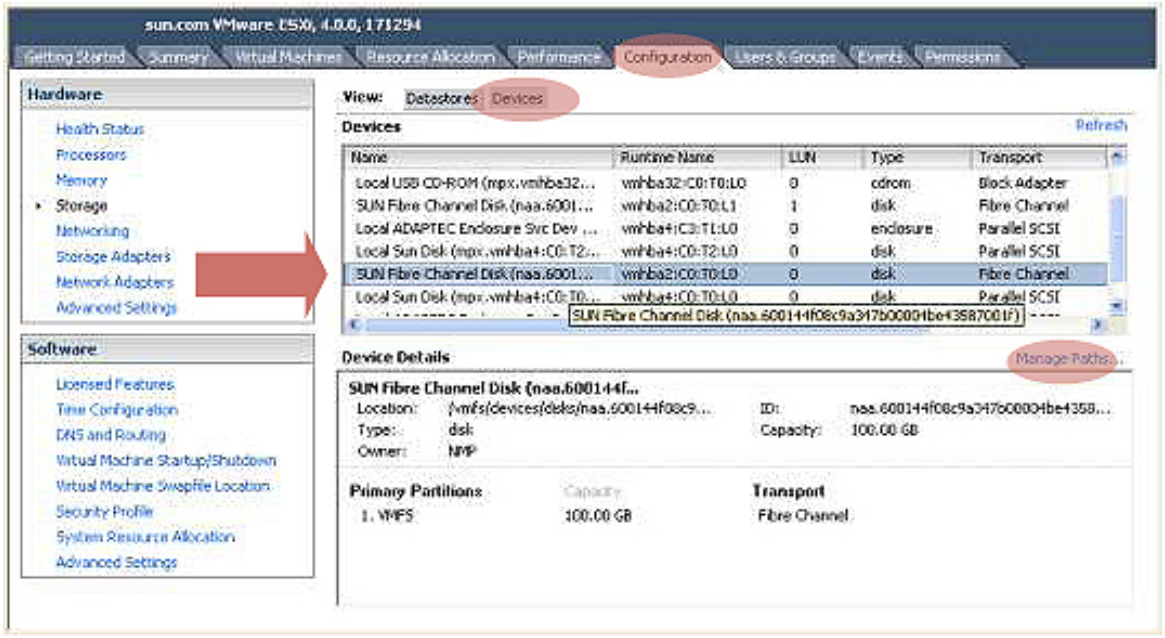

図29. VMware FC LUNデバイス情報

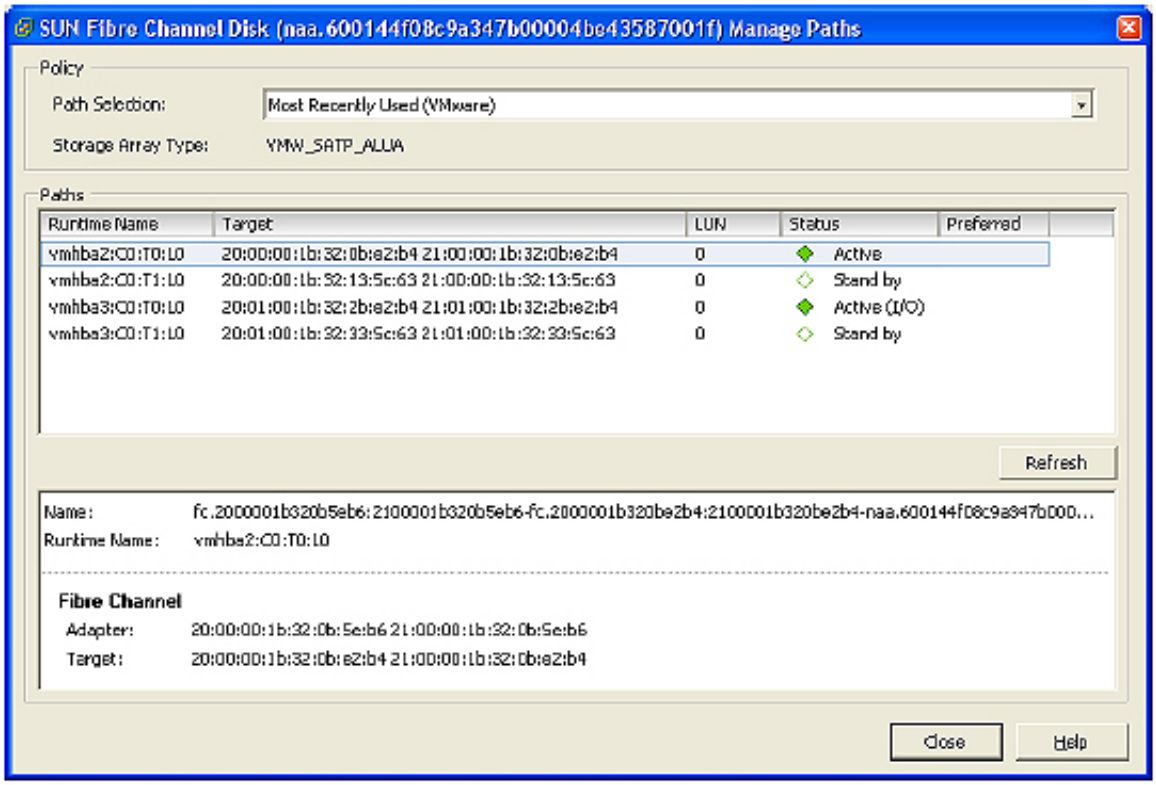

図30. VMware FC LUNパス情報

付録C：参考資料

参考文献

Sun ZFS Storage Appliance Documentation

ブログ

Oracle Blogsポータル

Aligning Flash Modules for Optimal Performance

A Quick Guide to the ZFS Intent Log (ZIL)

Brendan Gregg, ZFS L2ARC

ORACLE

Sun ZFS Storage Applianceにおけるファイバ・チャネルの使用について
2012年3月、バージョン1.0

著者：Peter Brouwer
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

お問い合わせ窓口

Oracle Direct
TEL 0120-155-096
URL oracle.com/jp/direct

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together