

Symantec™ Storage  
Foundation and High  
Availability Solutions 6.2  
SmartIO for Solid State  
Drives ソリューションガイド -  
Linux

# Symantec™ Storage Foundation Solutions SmartIO ソリューションガイド

この本で説明されているソフトウェアは使用許諾契約の下で提供され、同意条項に従う場合にのみ使うことができます。

製品のバージョン: 6.2

マニュアルバージョン: 6.2 Rev 1.

## 法的通知と登録商標

Copyright © 2015 Symantec Corporation. All rights reserved.

Symantec、Symantec ロゴ、Checkmark ロゴ、Veritas、Veritas Storage Foundation、CommandCentral、NetBackup、Enterprise Vault、LiveUpdate は、Symantec Corporation または同社の米国およびその他の国における関連会社の商標または登録商標です。その他の会社名、製品名は各社の登録商標または商標です。

本書に記載の製品は、ライセンスに基づいて配布され、使用、コピー、配布、逆コンパイル、リバースエンジニアリングはそのライセンスによって制限されます。本書のいかなる部分も、Symantec Corporation とそのライセンサーの書面による事前の許可なく、いかなる形式、方法であっても複製することはできません。

本書は「現状有姿のまま」提供され、商品性、特定目的への適合性、不侵害の黙示的な保証を含む、すべての明示的または黙示的な条件、表明、保証は、この免責が法的に無効であるとみなされないかぎり、免責されるものとします。Symantec Corporation は、本書の供給、性能、使用に関する付随的または間接的損害に対して責任を負わないものとします。本書に記載の情報は、予告なく変更される場合があります。

ライセンス対象ソフトウェアと関連書類は、FAR 12.212 の規定によって商用コンピュータソフトウェアとみなされ、Symantec により構内サービスとホスト型サービスのどちらとして提供されるかにかかわらず、場合に応じて、FAR 52.227-19「Commercial Computer Software - Restricted Rights」、DFARS 227.7202「Rights in Commercial Computer Software or Commercial Computer Software Documentation」、その後継規制の規定により制限された権利の対象となります。米国政府によるライセンス対象ソフトウェアと関連書類の使用、修正、複製のリリース、実演、表示または開示は、本使用許諾契約の条項に従ってのみ行われるものとします。

弊社製品に関して、当資料で明示的に禁止、あるいは否定されていない利用形態およびシステム構成などについて、これを包括的かつ暗黙的に保証するものではありません。また、弊社製品が稼動するシステムの整合性や処理性能に関しても、これを暗黙的に保証するものではありません。

これらの保証がない状態で、弊社製品の導入、稼動、展開した結果として直接的、あるいは間接的に発生した損害等についてこれが補償されることはありません。製品の導入、稼動、展開にあたっては、お客様の利用目的に合致することを事前に十分に検証および確認いただく前提で、計画および準備をお願いします。

Symantec Corporation  
350 Ellis Street  
Mountain View, CA 94043

<http://www.symantec.com>

# 目次

第 1 章	SFHA Solutions SmartIO の概要 .....	6
	ソリッドステートドライブの SmartIO について .....	6
	SFHA 環境の SmartIO について .....	7
	アクティブ/アクティブクラスタ環境の SmartIO について .....	8
	Linux 仮想化環境の SmartIO について .....	8
第 2 章	SmartIO 機能の使用: 使用例 .....	10
	VxVM ボリュームで動作しているアプリケーションの SmartIO 読み込み キャッシュについて .....	10
	VxVM ボリュームの SmartIO 読み取りキャッシュに必要な構成 .....	11
	VxVM ボリュームの自動キャッシュ .....	12
	VxVM ボリュームに対する SmartIO 読み取りキャッシュの設定 .....	13
	VxVM キャッシュ領域の検証とキャッシュの監視 .....	14
	VxFS ファイルシステムで実行しているアプリケーションの SmartIO 読み取 りキャッシュについて .....	16
	VxFS ファイルシステムの SmartIO 読み取りキャッシュに必要な構 成 .....	17
	VxFS ファイルシステムに対する自動キャッシュ .....	17
	VxFS ファイルシステムに対する SmartIO 読み取りキャッシュの設 定 .....	18
	VxFS キャッシュ領域の確認およびキャッシュの監視 .....	20
	キャッシュ動作のカスタマイズ .....	21
	VxFS ファイルシステムで実行しているアプリケーションの SmartIO ライト バックキャッシュについて .....	23
	VxFS ファイルシステムの SmartIO ライトバックキャッシュに必要な設 定 .....	24
	VxFS ファイルシステムの SmartIO ライトバックキャッシュの設定 .....	25
	VxFS キャッシュ領域の確認およびキャッシュの監視 (ライトバックモー ド) .....	26
	VxFS ファイルシステムの Oracle データベースの SmartIO キャッシュにつ いて .....	30
	SmartIO plug-in for Oracle を使うための前提条件と設定 .....	31
	VxFS ファイルシステムで動作しているデータベースに対するデフォ ルトの SmartIO キャッシュポリシーの設定 .....	31

	データベースオブジェクトに対する SmartIO キャッシュポリシーの設定 .....	36
	データベースオブジェクトのピン設定とピン解除 .....	36
	データベースに対するキャッシュの有効化と無効化 .....	37
	データベースに対するキャッシュポリシー詳細のリスト .....	38
	データベースに対するキャッシュ統計のリスト .....	38
	VxVM ボリュームのデータベースの SmartIO キャッシュについて .....	39
	VxVM ボリュームに対する SmartIO データベースキャッシュテンプレート レートの適用 .....	39
<b>第 3 章</b>	<b>SmartIO の管理</b> .....	<b>44</b>
	キャッシュ領域の作成 .....	44
	キャッシュ領域に関する情報の表示 .....	47
	データオブジェクトのキャッシュの有効化または無効化 .....	49
	ファイルシステムのキャッシュの有効化または無効化 .....	49
	データボリュームのキャッシュの有効化または無効化 .....	50
	キャッシュ領域へのデバイスの追加 .....	50
	ボリュームからキャッシュ領域へのキャッシュの一時停止 .....	51
	キャッシュ領域からのデバイスの削除 .....	51
	キャッシュ領域の破棄 .....	52
	VxVM キャッシュ領域の属性の設定 .....	53
	VxFS キャッシュ領域に対するキャッシュモードの設定または変更 .....	54
	ライトバックキャッシュ領域からのダーティデータのフラッシュ .....	55
	ライトバックキャッシュのチューニング .....	55
	SmartIO キャッシュ統計の表示 .....	57
	VxVM キャッシュエリアの詳細なキャッシュ統計の表示 .....	59
	VxFS キャッシュ領域に対する詳細キャッシュ統計の表示 .....	60
<b>第 4 章</b>	<b>トラブルシューティングとエラー処理</b> .....	<b>63</b>
	永続または「ウォーム」VxVM キャッシュのサポート .....	63
	古いキャッシュを持つプライマリボリュームのエラーによるデータ破損 の可能性 .....	64
	HA フェールオーバー時のキャッシュの移行はサポートされない .....	64
	キャッシュ領域がディスク障害後に失われる (3158482) .....	64
	再ブート後にキャッシュがオンラインにならない .....	65
	ノードエラー後のライトバックキャッシュのリカバリ .....	65
<b>付録 A</b>	<b>コマンドリファレンス</b> .....	<b>67</b>
	SmartIO コマンドリファレンス .....	67
<b>索引</b>	.....	<b>69</b>

# SFHA Solutions SmartIO の概要

この章では以下の項目について説明しています。

- ソリッドステートドライブの SmartIO について
- SFHA 環境の SmartIO について
- アクティブ/アクティブクラスタ環境の SmartIO について
- Linux 仮想化環境の SmartIO について

## ソリッドステートドライブの SmartIO について

ソリッドステートドライブ (SSD) は、回転ディスクを備えていないデバイスです。DRAM や NAND フラッシュなどの現在のソリッドステート技術は、従来の回転ディスクに比べ、より高速なデータアクセス、より優れた効率性、より小さな占有領域を実現しています。データセンターでは、サーバー内、すべてのフラッシュアレイ、すべてのフラッシュアプライアンス、従来の HDD アレイとの統合など、さまざまなフォームファクタでソリッドステート技術が使われます。それぞれのフォームファクタには異なる提案値が存在します。さらに、SSD には PCIe、FC、SATA、SAS といった豊富な接続タイプがあります。

SSD デバイスで 1 GB あたり必要な現在のコストにより、SSD の値が大容量のストレージデバイスの値として最適ではありません。SSD を採用するメリットはパフォーマンスを改善し、秒単位の I/O (IOPS) のコストを削減することです。データ効率と配置は、データセンターによるソリッドステートへの投資に対する収益を最大化させる重要な要素です。

SFHA Solutions (Storage Foundation and High Availability Solutions) の SmartIO 機能は、I/O キャッシュを介して SSD のデータ効率性を改善します。効率性を改善する SmartIO を使用して、IOP ごとのコストを最適化できます。SmartIO は、その基盤となるハードウェア技術の深い知識を必要としません。SmartIO は、高度でカスタマイズ可能なヒューリスティックを使って、キャッシュに保存するデータ、そのデータをキャッシュから削

除する方法を決定します。このヒューリスティックでは、ワークフローの特性に関する SFHA Solutions の知識が活用されます。

SmartIO はターゲットデバイスまたはデバイスのキャッシュ領域を使います。キャッシュ領域は、SmartIO がキャッシュしたデータとそのデータに関するメタデータを格納するために使うストレージ領域です。キャッシュ領域のタイプに応じて、VxFS キャッシュまたは VxVM キャッシュのいずれかがサポートされるかが決定します。SmartIO を使い始めるために、アプリケーションのオンライン時に単一のコマンドでキャッシュ領域を作成できます。

アプリケーションが I/O 要求を発行すると、SmartIO はキャッシュからその I/O に対応できるかどうかを確認します。アプリケーションが下位のボリュームまたはファイルシステムからデータにアクセスすると、内部ヒューリスティックに基づき特定のデータがキャッシュに移動します。後続の I/O はキャッシュから処理されます。

SmartIO は、VxVM ボリュームにマウントされた VxFS ファイルシステムの読み取りと書き込みキャッシュを複数のキャッシュモードと設定でサポートします。SmartIO は、VxVM ボリュームで動作するアプリケーションのブロックレベル読み取りキャッシュもサポートします。

p.10 の「[VxVM ボリュームで動作しているアプリケーションの SmartIO 読み込みキャッシュについて](#)」を参照してください。

p.16 の「[VxFS ファイルシステムで実行しているアプリケーションの SmartIO 読み取りキャッシュについて](#)」を参照してください。

p.23 の「[VxFS ファイルシステムで実行しているアプリケーションの SmartIO ライトバックキャッシュについて](#)」を参照してください。

p.30 の「[VxFS ファイルシステムの Oracle データベースの SmartIO キャッシュについて](#)」を参照してください。

p.39 の「[VxVM ボリュームのデータベースの SmartIO キャッシュについて](#)」を参照してください。

## SFHA 環境の SmartIO について

クラスタ化環境では、SmartIO のキャッシュはクラスタの各ノードのローカルに保存されます。キャッシュ領域は 1 つのノードからオフラインにできません。別のノードでオンラインにすることもできません。

SFHA を使う場合のようにアクティブ/パッシブ環境で SmartIO 読み込みキャッシュを使う場合は、別のノードにデータボリュームやファイルシステムをデポートまたはインポートできます。SmartIO のキャッシュは他のノードに移動されません。ローカルの SmartIO キャッシュ領域で新しいノードのキャッシュを開始します。SmartIO はフェールオーバー時に高可用性のクラスタのキャッシュを移行するための機能を提供しません。

VxFS ライトバックキャッシュでは、ディスクグループのフェールオーバーは推奨しません。ローカルマウントの場合にはキャッシュをミラー化しないと、ノードに失敗した場合のファイ

ルのディスクコピーが不完全または古くなる可能性があります。代わりに、キャッシュのデータフレクシオンを提供する SFCFSHA を使うことを推奨します。

## アクティブ/アクティブクラスタ環境の SmartIO について

SFRAC (Symantec Storage Foundation for Oracle RAC) などのアクティブ/アクティブ環境で SmartIO 読み込みキャッシュを使うと、各ノードのローカルの SmartIO キャッシュ領域ですべてのノードのキャッシュを開始します。キャッシュ領域は各ノードで排他的にオンラインまたはオフラインにできます。

SmartIO VxFS のライトバックキャッシュは現在 SFRAC でサポートされていません。

## Linux 仮想化環境の SmartIO について

Linux 仮想化環境でゲストに Storage Foundation and High Availability Solutions をインストールすると、SmartIO を使って SSD または他のサポート対象の高速デバイスでデータをキャッシュに保存できます。

KVM 環境や RHEV 環境では、SmartIO キャッシュはゲストのライブ移行をサポートしません。

VMware の場合は、ESXi ハイパーバイザで DMP for VMware (SmartPool) が有効になっていると SmartIO は vMotion をサポートします。

Storage Foundation for Oracle RAC は LINUX の仮想化環境でサポートされません。

次の表に、Linux 仮想化環境で SmartIO を使う方法を示します。

表 1-1 は KVM 環境での SmartIO の使い方を示します。

表 1-1 Linux: KVM での SmartIO のサポート

ゲストでの設定:	ホストでの設定:	キャッシュの実行:	VxVM 読み込みキャッシュ	VxFS 読み込みキャッシュ	VxFS ライトバックキャッシュ
SF	任意 (SF または SFCFSHA)	ゲストでの	はい	はい	はい
SFHA	任意 (SF または SFCFSHA)	ゲストでの	はい	はい	はい
SFCFSHA	任意 (SF または SFCFSHA)	ゲストでの	はい	はい	はい
指定なし (Any)	SF	ホストでの	はい	はい	はい
指定なし (Any)	SFCFSHA	ホストでの	はい	はい	はい



表 1-2 では RHEV 環境での SmartIO の使い方を示します。

表 1-2 Linux: RHEV での SmartIO のサポート

ゲストでの設定:	ホストでの設定:	キャッシュの実行:	VxVM 読み込みキャッシュ	VxFS 読み込みキャッシュ	VxFS ライトバックキャッシュ
SF	任意 (SF または SFCFSHA)	ゲストでの	はい	はい	はい
SFHA	任意 (SF または SFCFSHA)	ゲストでの	はい	はい	はい
SFCFSHA	任意 (SF または SFCFSHA)	ゲストでの	はい	はい	はい
指定なし (Any)	SF	ホストでの	はい	はい	はい
指定なし (Any)	SFCFSHA	ホストでの	はい	はい	はい

表 1-3 では VMware 環境での SmartIO の使い方を示します。

表 1-3 Linux: VMware での SmartIO のサポート

ゲストでの設定:	ホストでの設定:	キャッシュの実行:	VxVM 読み込みキャッシュ	VxFS 読み込みキャッシュ	VxFS ライトバックキャッシュ
SF	DMP for VMware (省略可能)	ゲストでの	はい	はい	いいえ
SFHA	DMP for VMware (省略可能)	ゲストでの	はい	はい	いいえ
SFCFSHA	DMP for VMware (省略可能)	ゲストでの	はい	はい	いいえ

Linux の仮想化環境での Storage Foundation and High Availability Solutions の設定について詳しくは『Symantec Storage Foundation and High Availability Solutions 仮想化ガイド - Linux』を参照してください。

# SmartIO 機能の使用: 使用例

この章では以下の項目について説明しています。

- **VxVM** ボリュームで動作しているアプリケーションの **SmartIO** 読み込みキャッシュについて
- **VxFS** ファイルシステムで実行しているアプリケーションの **SmartIO** 読み取りキャッシュについて
- **VxFS** ファイルシステムで実行しているアプリケーションの **SmartIO** ライトバックキャッシュについて
- **VxFS** ファイルシステムの **Oracle** データベースの **SmartIO** キャッシュについて
- **VxVM** ボリュームのデータベースの **SmartIO** キャッシュについて

## VxVM ボリュームで動作しているアプリケーションの SmartIO 読み込みキャッシュについて

SmartIO は、Veritas Volume Manager (VxVM) ボリュームの遮断レベルの読み込みキャッシュをサポートします。このタイプの **SmartIO** キャッシュは、第一に未加工のボリュームで直接動作するアプリケーションをサポートします。たとえば、未加工のボリュームで直接動作するデータベースインスタンスをサポートします。**VxFS** キャッシュを使うことができない場合に、ボリュームレベルのキャッシュを使うこともできます。**SmartIO** は、ボリュームレベルの読み込みキャッシュのみをサポートします。

**SmartIO** キャッシュは通常、1 つ以上の **SSD** デバイスや他の高速デバイスに存在します。**SmartIO** は、アプリケーション読み込み I/O を標準ストレージではなく **SSD** ベースキャッシュから処理するので読み込み I/O の処理速度を上げます。

SmartIO ではキャッシュの設定に複雑な設定を行う必要はありません。キャッシュに保存したデータとキャッシュに関するメタデータのストレージ領域であるキャッシュ領域を単に設定するだけです。ボリュームレベルの読み込みキャッシュの場合は、VxVM タイプのキャッシュ領域があります。システムごとに単一の VxVM キャッシュ領域を使います。デフォルトでは、SmartIO のキャッシュ領域はシステムのすべての VxVM ボリュームに対して自動キャッシュを有効にします。必要に応じて、キャッシュ領域を `noauto` に設定できます。noauto キャッシュ領域の場合は、VxVM ボリュームの SmartIO 読み込みキャッシュを明示的に有効にする必要があります。キャッシュ領域の設定は永続的です。

p.12 の「[VxVM ボリュームの自動キャッシュ](#)」を参照してください。

キャッシュが有効な各 VxVM ボリュームでは、SmartIO はキャッシュに保存するデータやキャッシュから退去させるデータを判断します。SmartIO は、作業負荷の知識を使ってキャッシュの使用を最適化します。

SmartIO 機能は、1 つのシステムで 1 つの VxVM キャッシュ領域のみをサポートします。各システムで、キャッシュに保存されたすべての VxVM ボリュームは VxVM タイプの単一キャッシュ領域を共有します。同一システムに VxFS キャッシュ領域と VxVM キャッシュ領域の両方を設定できますが、複数の VxVM キャッシュ領域はサポートされません。

キャッシュ領域はクラスタの各ノード専用です。キャッシュの内容はクラスタのノード全体では共有されません。

SmartIO のキャッシュは、ボリュームレベルでキャッシュの一貫性を保ちます。キャッシュが有効な場合にキャッシュデバイスがアクセス不能になっても、アプリケーションは正常に機能し続けます。ただし、アプリケーションのパフォーマンスは低下することがあります。

CVM (Cluster Volume Manager) 環境では、SmartIO は共有ボリュームに書き込むときにキャッシュの一貫性プロトコルを使って複数ノードのキャッシュ領域の一貫性を保ちます。データボリュームに書き込むと他のノードのキャッシュ領域の内容が無効になります。キャッシュの一貫性プロトコルは GLM (Group Lock Manager) モジュールを使って通信します。キャッシュを初めて受信するときに、書き込み I/O パスでキャッシュの一貫性プロトコルのパフォーマンスに小さいオーバーヘッドが起きます。

読み込みキャッシュのデータはデフォルトでは永続的ではありません。計画済みのシステム再ブートの場合は、ウォームキャッシュの作成を選択できます。

p.63 の「[永続または「ウォーム」VxVM キャッシュのサポート](#)」を参照してください。

## VxVM ボリュームの SmartIO 読み取りキャッシュに必要な構成

次の構成により、VxVM ボリュームの読み取りキャッシュ用に SmartIO を設定できます。

- SFRAC (Storage Foundation RAC) クラスタまたは SFCFSHA (Storage Foundation Cluster File System High Availability) クラスタ。キャッシュ領域は共有ボリューム上に存在することができません。キャッシュ領域の共有アクセスはサポートされていないため、VxVM キャッシュ領域は各ノードでローカルに設定する必要があります。

- SFHA (Storage Foundation High Availability) クラスタ キャッシュ領域の共有アクセスはサポートされていないため、VxVM キャッシュ領域は各ノードでローカルに設定する必要があります。  
p.7 の「SFHA 環境の SmartIO について」を参照してください。
- スタンドアロン Storage Foundation システム  
キャッシュされるボリュームには、190 以上のディスクグループバージョンが必要です。  
キャッシュ領域に使うデバイスには次の特性があります。
- VxVM (Veritas Volume Manager) がサポートするソリッドステートドライブ (SSD) など高速なデバイスをつかかって、読み込み IO パフォーマンスを向上させます。ただし、VxVM によってサポートされる任意のデバイスをキャッシュ領域に使うことができます。
- VxVM 用に初期化し、cddisk 形式である必要があります。

## VxVM ボリュームの自動キャッシュ

キャッシュ領域の関連付けタイプは、システムの自動キャッシュを有効にするかどうかを示します。VxVM キャッシュの関連付けタイプの属性は永続的です。関連付けタイプは、次のいずれかです。

- auto 属性 (デフォルト)  
キャッシュ領域の自動キャッシュを有効にします。VxVM ボリュームのキャッシュを明示的に無効にしないかぎり、システムの VxVM データボリュームはすべてキャッシュに保存されます。明示的にボリュームのキャッシュを有効にする必要はありません。  
SmartIO は RAID-5 ボリュームと DCO ボリュームのキャッシュをサポートしません。  
SmartIO では、SRL (Storage Replication Log)、DCM (Data Change Map) などのログやキャッシュオブジェクトに使うボリュームや、領域最適化スナップショットのキャッシュオブジェクトに使うボリュームを自動的にキャッシュに保存することはできません。  
デフォルトでは、VxVM キャッシュ領域に auto 属性が指定されます。
- noauto 属性  
キャッシュ領域の自動キャッシュを行うことはできません。ボリュームを自動的にキャッシュに保存しません。キャッシュに保存したいボリュームそれぞれのキャッシュを明示的に有効にする必要があります。以前有効だったボリュームを除外する場合以外は、キャッシュに保存するボリュームを明示的に無効にする必要はありません。ボリュームを作成するとキャッシュを有効にすることができます。I/O を静止せずに既存の VxVM ボリュームで読み込みキャッシュの有効化または無効化を選択することもできます。

## VxVM ボリュームに対する SmartIO 読み取りキャッシュの設定

読み取りモードでは、SmartIO 機能により VxVM I/O がキャッシュに格納されます。VxVM ボリュームに対して読み取りキャッシュを行うための SmartIO を設定するには、キャッシュ領域を作成するだけです。

### VxVM ボリュームに対する SmartIO 読み取りキャッシュの設定

- 1 次のコマンドのいずれかを使用して、SSD デバイスで VxVM タイプのキャッシュ領域を作成します。

- デバイスのディスクアクセス名 (*daname*) を使って 1 つ以上のデバイスを指定します。デバイスは VxVM 用に初期化し、*cdsdisk* 形式である必要があります。

```
# sfcache create -t VxVM [size] daname[...] ¥  
[cacheline_size=cacheline_size] [--auto|--noauto] [--nostripe|ncols=M]
```

ここで、

*daname* は、キャッシュ領域を作成するデバイスのディスクアクセス名を指定します。

*cachearea\_name* は、キャッシュ領域のカスタマイズした名前です。指定しない場合、SmartIO 機能によりキャッシュ領域の名前が自動的に生成されます。

*size* はキャッシュ領域のサイズを指定します。デフォルトでは、デバイスで利用可能な容量のすべてがキャッシュ領域に使われます。

*--noauto* | *--auto* はキャッシュ領域のタイプを指定します。デフォルトは *--auto* です。

*--nostripe* | *ncols=n* はキャッシュ領域のレイアウトオプションを指定します。デフォルトでは、複数のディスクを指定するとキャッシュ領域はストライプボリューム上に作成されます。 *ncols=n* オプションを使ってストライプボリュームの列数を指定します。指定したディスクの連結ボリュームにキャッシュ領域を作成するには *--nostripe* オプションを使います。

*cacheline\_size* は、SmartIO がキャッシュに使う単位を指定します。アプリケーション I/O がデータにアクセスするとき、SmartIO は *cacheline* のサイズに基づきそのデータをキャッシュに移動します。通常、*cacheline\_size* を変更する必要はありません。

次に例を示します。

```
# sfcache create -t VxVM ssd0_0
```

- キャッシュ領域に使うディスクグループとボリュームの名前を指定することもできます。この場合、サイズは指定できません。ボリューム全体がキャッシュ領域に使われます。

```
# sfcache create -t VxVM [cacheline_size=cacheline_size] ¥  
[--noauto|--auto] dg/vol
```

ここで、

`dg/vol` は、キャッシュ領域に使うディスクグループ名とボリューム名を指定します。  
`SmartIO` はキャッシュ領域にこの名前を使います。

`--noauto` | `--auto` は、キャッシュ領域の関連付けタイプを指定します。デフォルトは `--auto` です。

次に例を示します。

```
# sfcache create -t VxVM --auto ssd_dg/ssd_vol
```

- 2 必要な VxVM ボリュームに対しキャッシュを有効にします。キャッシュ領域が自動の場合、この手順は不要です。キャッシュは、SSD デバイスがオンラインになるときにすべての VxVM ボリュームに対しデフォルトで有効になります。

キャッシュ領域が**非自動**の場合、キャッシュに格納されるボリュームに対しキャッシュを有効にする必要があります。

```
# sfcache enable [--read] dg/vol
```

ここで、

`dg/vol` により、キャッシュに格納するボリュームのディスクグループ名とボリューム名が指定されます。

次に例を示します。

```
# sfcache enable mydg/vol1
```

## VxVM キャッシュ領域の検証とキャッシュの監視

SmartIO 機能を設定すると、キャッシュ領域が存在すること、キャッシュが発生していることを検証できます。

### キャッシュ領域の検証と監視を行うには

- 1 システム上のキャッシュ領域に関する情報を表示するには、次のコマンドを使います。

```
# sfcache list -l

Cachearea: sfcachearea_1
Assoc Type: AUTO
Type: VxVM
Size: 30.00g
Cacheline Size: 64.00k
Memory Size: 16.00m
State: ONLINE
Layout: CONCAT
Number of Columns: 0

ASSOCIATED DATA OBJECTS:

Volume: testdg/testvol1
Size: 500.00g
State: ENABLED
Kstate: ENABLED
Caching Mode: read

Volume: testdg/testvol2
Size: 500.00g
State: ENABLED
Kstate: ENABLED
Caching Mode: read
```

p.47 の「[キャッシュ領域に関する情報の表示](#)」を参照してください。

- 2 特定のキャッシュ領域に関する情報を表示するには、次のコマンドを使います。

```
# sfcache list sfcachearea_1
Cachearea: sfcachearea_1
Assoc Type: AUTO
Type: VxVM
Size: 30.00g
Cacheline Size: 64.00k
Memory Size: 16.00m
State: ONLINE
Layout: CONCAT
Number of Columns: 0
```

ASSOCIATED DATA OBJECTS:

ASSOC DATAOBJECT NAME	CACHING-MODE	STATE	KSTATE
testdg/testvol1	read	ENABLED	ENABLED
testdg/testvol2	read	ENABLED	ENABLED

p.57 の「[SmartIO キャッシュ統計の表示](#)」を参照してください。

- 3 キャッシュ使用率の統計を表示するには、次のコマンドを使います。

```
# sfcache stat sfcachearea_1
```

NAME	HIT RATIO		ART (Hit)ms		ART (Miss)ms		BYTES		
	%CACHE	RD	WR	RD	WR	RD	WR	RD	WR
TYPE: VxVM									
sfcachearea_1	13.43	91.24	94.20	0.142	0.819	0.414	0.798	15.31g	4.21g

ASSOCIATED DATA OBJECTS:

testdg/testvol1	6.10	90.00	96.00	0.141	0.459	0.348	0.448	6.77g	1.89g
testdg/testvol2	7.32	91.00	92.00	0.143	1.179	0.480	1.149	8.54g	2.31g

## VxFS ファイルシステムで実行しているアプリケーションの SmartIO 読み取りキャッシュについて

Storage Foundation and High Availability Solutions は、VxFS ファイルシステムで動作しているアプリケーションのソリッドステートドライブ (SSD) で読み込みキャッシュをサポートします。このシナリオでは、アプリケーションの読み込みは可能なかぎりキャッシュから行われます。アプリケーションはファイルシステムにアクセスするので、ファイルシステムはディスクのデータをキャッシュにロードします。アプリケーションの書き込みは通常の方法でディスクに対して行われます。書き込むたびに、アプリケーションが古いデータ



を参照しないようにファイルシステムはキャッシュを同期します。キャッシュデバイスが壊れても、読み込みモードでキャッシュに保存されたファイルはディスクに完全に存在します。したがって、キャッシュのエラーはファイルのアプリケーション I/O に影響せず、アプリケーション I/O は中断せずに続行します。

## VxFS ファイルシステムの SmartIO 読み取りキャッシュに必要な構成

次の構成により、VxFS ファイルシステムの読み取りキャッシュ用に SmartIO を設定できます。

- 各ノードにプライベート SSD を備えた SFRAC (Storage Foundation RAC) クラスタ、SFCFSHA (Storage Foundation Cluster File System High Availability)、SFHA (Storage Foundation High Availability) クラスタ。  
p.7 の「[SFHA 環境の SmartIO について](#)」を参照してください。

- スタンドアロンの Storage Foundation システム。

ファイルシステムは次の特性を備える必要があります。

- ファイルシステムレイアウトバージョン 10。
- VxVM ボリュームにマウントする必要があります。

## VxFS ファイルシステムに対する自動キャッシュ

SmartIO 機能はシステムの 1 つの VxFS キャッシュ領域のみをサポートします。システムごとに、キャッシュに格納されるすべての VxFS ファイルシステムで VxFS タイプの単一のキャッシュ領域が共有されます。

キャッシュ領域の関連付けタイプは、システムの自動キャッシュを有効にするかどうかを示します。VxFS キャッシュ領域に対する関連タイプ属性は、永続的です。関連付けタイプは、次のいずれかです。

- auto 属性 (デフォルト)  
キャッシュ領域の自動キャッシュを有効にします。システムのすべてのファイルシステムは、そのファイルシステムに対しキャッシュを明示的に無効にしないかぎり、キャッシュに格納されます。ファイルシステムでキャッシュを明示的に有効にする必要はありません。デフォルトで、VxFS キャッシュ領域には auto 属性があります。
- noauto 属性  
キャッシュ領域の自動キャッシュを行うことはできません。ファイルシステムは自動的にキャッシュに格納されません。キャッシュに格納するファイルシステムごとにキャッシュを明示的に有効にする必要があります。前に有効にされたファイルシステムを除き、ファイルシステムのキャッシュを明示的に無効にする必要はありません。

キャッシュ領域はクラスタの各ノード専用です。クラスタファイルシステムの場合、クラスタの各ノードには独自のキャッシュ領域があります。キャッシュはノード単位で発生し、キャッ

シユ内容はクラスタのノード間で共有されることはありません。キャッシュが有効になっているファイルシステムは、各ノードのローカルキャッシュ領域と関連付けられています。

## VxFS ファイルシステムに対する SmartIO 読み取りキャッシュの設定

読み取りモードでは、SmartIO 機能により VxFS ファイルシステム読み取り I/O がキャッシュに格納されます。VxFS ファイルシステムに対して読み取りキャッシュを行うための SmartIO を設定するには、キャッシュ領域を作成するだけです。

### VxFS ファイルシステムに対する SmartIO 読み取りキャッシュの設定

- 1 次のコマンドのいずれを使って、SSD デバイスで VxFS キャッシュ領域を作成します。

- デバイスのディスクアクセス名 (**daname**) を使って 1 つ以上のデバイスを指定します。デバイスは VxVM 用に初期化し、`cdsdisk` 形式である必要があります。

```
# sfcache create [-t VxFS] [size] daname[...] [--auto|--noauto] ¥  
  [--nostripe|ncols=n] [cachearea_name]
```

ここで、

**daname** は、キャッシュ領域を作成するデバイスのディスクアクセス名を指定します。

**cachearea\_name** は、キャッシュ領域のカスタマイズした名前です。指定しない場合、SmartIO 機能によりキャッシュ領域の名前が自動的に生成されます。

**size** はキャッシュ領域のサイズを指定します。デフォルトでは、デバイスで利用可能な容量のすべてがキャッシュ領域に使われます。

`--nostripe|ncols=n` はキャッシュ領域のレイアウトオプションを指定します。デフォルトでは、複数のディスクを指定するとキャッシュ領域はストライプボリューム上に作成されます。`ncols=n` オプションを使ってストライプボリュームの列数を指定します。指定したディスクの連結ボリュームにキャッシュ領域を作成するには `--nostripe` オプションを使います。

`--noauto|--auto` はキャッシュ領域のタイプを指定します。デフォルトは `--auto` です。

次に例を示します。

```
# sfcache create ssd0_0
```

- キャッシュ領域に使うディスクグループとボリュームの名前を指定することもできます。この場合、サイズは指定できません。ボリューム全体がキャッシュ領域に使われます。

```
# sfcache create [-t VxFS] [--noauto|--auto] dg/vol
```

ここで、

`dg/vol` は、キャッシュ領域に使うディスクグループ名とボリューム名を指定します。  
SmartIO はキャッシュ領域にこの名前を使います。

`--noauto` | `--auto` はキャッシュ領域のタイプを指定します。デフォルトは `--auto` です。

次に例を示します。

```
# sfcache create --auto ssd_dg/ssd_vol
```

## 2 ファイルシステムがマウントされていない場合、VxFS ファイルシステムをマウントします。

- キャッシュ領域が自動の場合、読み取りキャッシュは VxFS ファイルシステムのマウント時に有効になります。

たとえば、ローカルマウントの場合:

```
# mount -t vxfs /dev/vx/dsk/testdg/voll /mnt1
```

たとえば、CFS マウントの場合:

```
# mount -t vxfs -o cluster /dev/vx/dsk/testdg/voll /mnt1
```

- キャッシュ領域が非自動の場合、キャッシュに格納する各 VxFS ファイルシステムに対しキャッシュを有効にする必要があります。キャッシュを有効にするには、ファイルシステムを `-o smartiomode` オプションでマウントします。

たとえば、ローカルマウントの場合:

```
# mount -t vxfs -o smartiomode=read /dev/vx/dsk/testdg/voll /mnt1
```

たとえば、CFS マウントの場合:

```
# mount -t vxfs -o cluster,smartiomode=read /dev/vx/dsk/testdg/voll /mnt1
```

ファイルシステムがマウントされた後にキャッシュを有効にすることもできます。

```
# sfcache enable mount_point
```

ここで、

**`mount_point`** はファイルシステムのマウントポイントです。

次に例を示します。

```
# sfcache enable /mnt1
```

## 3 必要に応じて、キャッシュの動作をさらにカスタマイズすることができます。

p.21 の「[キャッシュ動作のカスタマイズ](#)」を参照してください。

## VxFS キャッシュ領域の確認およびキャッシュの監視

SmartIO 機能を設定すると、キャッシュ領域が存在すること、キャッシュが発生していることを検証できます。

VxFS のキャッシュ領域では、`sfcache list` コマンドはファイルやディレクトリのキャッシュモードを示します。モードを明示的に設定しないと、ファイルやディレクトリはマウントポイントのキャッシュモードを継承します。ファイルやディレクトリにモードを明示的に設定すると、その値をすべての再マウントで継続して使います。表示されるキャッシュモードは、マウントポイントで有効なモードとは異なることがあります。writeback モードは、ファイルシステムを writeback モードでマウントしなければ有効になりません。ファイルやディレクトリを writeback モードに設定してもファイルシステムを別のモードでマウントしている場合には、ファイルやディレクトリはマウントポイントのキャッシュモードを継承します。

キャッシュ領域の検証と監視を行うには

- 1 システム上のキャッシュ領域に関する情報を表示するには、次のコマンドを使います。

```
# sfcache list

NAME                TYPE SIZE  ASSOC-TYPE  STATE  DEVICE
sfcachearea_2      VxFS 31.97g AUTO        ONLINE  ssd0_0
```

- 2 特定のキャッシュ領域に関する情報を表示するには、次のコマンドを使います。

```
# sfcache list sfcachearea_2
Cachearea: sfcachearea_2
Assoc Type: AUTO
Type: VxFS
Size: 9.96g
State: ONLINE
Layout: CONCAT
Number of Columns: 0

/dev/vx/dsk/sfcache_defaultdg/sfcachearea_2:
FSUUID                                SIZE  MODE  MOUNTPOINT
3671ff51-9696-0000-872c-000056bcb408 39.0 MB  read  /mnt1
```

3 特定のファイルシステムに関する情報を表示するには

```
# sfcache list /mnt1

/mnt1:
READ CACHE      WRITEBACK      MODE      PINNED      NAME
    39.0 MB          0 KB      read      yes      /mnt1/dir
    39.0 MB          0 KB      read      yes      /mnt1
```

4 キャッシュ使用率の統計を表示するには、次のコマンドを使います。

```
# sfcache stat /mnt1

Cache Size:      9.97 GB
Cache Utilization:  39.0 MB ( 0.38 %)

Read Cache
Hit Ratio      Data Read      Data Written

/mnt2:
    0.00 %          0 KB          39.0 MB
```

出力には、キャッシュに格納されたデータに対する統計が表示されます。

p.57 の「[SmartIO キャッシュ統計の表示](#)」を参照してください。

## キャッシュ動作のカスタマイズ

デフォルトでは、SmartIO によりファイルデータが作業負荷に基づいてキャッシュに格納されます。SmartIO によりファイルの一部が I/O アクセスに基づいてキャッシュにロードされます。キャッシュ領域がいっぱいになると、新しいデータをキャッシュに格納するための領域を用意するためにデータが退去されることがあります。SmartIO では、データを退去するためにアクセスの頻度などの基準が使用されます。データがキャッシュにある間、そのファイルデータへの以降の I/O はキャッシュで受け入れられます。データが退去される場合、以降の I/O 要求はすべてプライマリストレージから処理されます。次に、SmartIO によりそのデータが再度キャッシュに格納される場合があります。

キャッシュの使用を最大化するためには、キャッシュ動作をカスタマイズして、ファイルがいつロードされるか、またはキャッシュから退去されるかを制御することができます。次の操作を使用して、キャッシュの動作をカスタマイズすることができます。

- load 操作は、I/O がファイルにアクセスする前にファイルをキャッシュにプリロードします。I/O がより迅速に戻るよう、ファイルはすでにキャッシュ内にあります。デフォルトで、ファイルはバックグラウンドでロードされます。-o sync 操作を使用して、ファイルを同期的にロードします。これにより、すべてのファイルがロードされるまでコマンドは戻りません。このようにロードされるファイルは、通常の退去基準の対象となります。

- pin 操作は、ファイルがキャッシュから退去されないようにします。一般的に使用されるファイルをピンで固定して SmartIO がファイルを退去しないようにし、後でそのファイルのキャッシュが再び必要になることを避けることができます。ピン設定されたファイルは、削除または明示的にピン解除されるまで無制限にキャッシュに保持されます。
  - o load オプションでファイルをピン設定すると、その操作によりファイル内容が同期的にキャッシュに格納されます。
  - o load オプションを指定しない場合、ファイル内容は I/O アクセスに基づいてキャッシュに格納されます。
- unpin 操作は、ピン設定された状態からファイルを解除します。unpin 操作によりファイルはすぐには退去されません。SmartIO では、キャッシュに領域が必要な場合、その他のファイルと同様にファイルの退去が考慮されます。

これらの操作ごとに、ファイルを個別に指定するか、またはディレクトリのすべてのファイルに影響するディレクトリ名を指定することができます。-r オプションを使用して、選択を再帰的に行います。

#### ファイルやディレクトリをロードするには

- ◆ キャッシュにファイルやディレクトリをロードするには、次のコマンドで 1 つ以上のファイル名またはディレクトリ名を指定します。

```
# sfcache load [-r] [-o sync] {file|dir}[file2|dir2...]
```

-r オプションを使用して、選択を再帰的に行います。

-o sync オプションを使用して、すべてのファイルがロードされるまでコマンドが戻らないように指定します。

#### ファイルやディレクトリをピン設定するには

- ◆ キャッシュにファイルやディレクトリをピン設定するには、次のコマンドで 1 つ以上のファイル名またはディレクトリ名を指定します。

```
# sfcache pin [-o load] [-r] {file|dir}[file2|dir2...]
```

-r オプションを使用して、選択を再帰的に行います。

-o load オプションを使用して、ファイルをキャッシュに同期的にロードします。

#### ファイルやディレクトリをピン解除するには

- ◆ キャッシュからファイルやディレクトリをピン解除するには、次のコマンドで 1 つ以上のファイル名またはディレクトリ名を指定します。

```
# sfcache unpin [-r] {file|dir} [file2|dir2...]
```

-r オプションを使用して、選択を再帰的に行います。

## VxFS ファイルシステムで実行しているアプリケーションの SmartIO ライトバックキャッシュについて

Storage Foundation and High Availability Solutions は、VxFS (Veritas File System) ファイルシステムで実行しているアプリケーションのソリッドステートドライブ (SSD) でライトバックキャッシュをサポートします。このシナリオでは、アプリケーションの読み込みと書き込みは可能なかぎりキャッシュから行われます。

---

**メモ:** SmartIO ライトバックキャッシュは、SF Oracle RAC 環境では現在サポートされません。

---

SmartIO を使うと `writeback` モードで書き込みキャッシュを行うことができます。`writeback` モードでは、通常は SSD にある SmartIO キャッシュにデータが書き込まれるとアプリケーションの書き込みの成功が返ります。後で、SmartIO はキャッシュをフラッシュしてディスクにダーティデータを書き込みます。ライトバックキャッシュは、ユーザーデータの同期書き込みの遅延を改善します。ディスクにダーティデータをフラッシュするときに書き込み順序が順守されないことがあります。

ライトバックキャッシュは読み込みキャッシュのスーパーセットです。ライトバックキャッシュが有効な場合は、読み込みキャッシュは暗黙に有効になります。可能であれば読み込みはキャッシュから行われ、ファイルシステムはキャッシュにファイルデータを透過的にロードします。読み込みキャッシュとライトバックキャッシュの両方が同一のファイルで同時に有効になることがあります。

`writeback` キャッシュモードにすると書き込み処理速度は速くなりますが、ディスクコピーが常に最新であるとはかぎりません。キャッシュデバイスが失敗した場合は、`writeback` モードでキャッシュに保存したファイルがディスクに一部しか存在しないことがあります。SmartIO には、デバイスがオンラインに戻ったときにキャッシュデバイスからデータをフラッシュする機構があります。SFCFSHA (Storage Foundation Cluster File System High Availability) をインストールすると、キャッシュリフレクションを使ってデータ損失から保護することもできます。

SFCFSHA では、`writeback` モードのキャッシュを有効にすると、SmartIO はファイルシステムレベルでライトバックデータを他のノードの SSD キャッシュにミラー化します。キャッシュリフレクションと呼ばれるこの動作は、ノードが失敗した場合にライトバックデータの消失を防ぎます。ノードが失敗した場合には、他のノードは損失ノードのミラー化ダーティデータを再設定の一環としてフラッシュします。キャッシュリフレクションでは、保留中のダーティデータが原因でノードが失敗してもライトバックデータは失われません。

ローカルマウントの場合にはキャッシュをミラー化しないと、ノードに失敗した場合のファイルのディスクコピーが不完全または古くなる可能性があります。

マウントポイントでライトバックキャッシュを有効にすると、そのファイルシステムで条件に合う同期書き込みをキャッシュに保存します。SmartIO は、次のような基準を使って書き込みがライトバックキャッシュの条件に合うかどうかを判断します。

- 書き込み要求を PAGESIZE で調整する必要がある(4 K の倍数)
- 書き込み要求が 2 MB 以下
- 書き込むファイルがマップされていない
- 管理者が writeback モードのキャッシュを明示的に無効にしていない

これらの判断を行う SmartIO 機能に役立つ情報を追加して、キャッシュに保存するデータをカスタマイズすることもできます。

## VxFS ファイルシステムの SmartIO ライトバックキャッシュに必要な設定

Storage Foundation または Storage Foundation High Availability のライトバックキャッシュで SmartIO を使うには Enterprise ライセンスが必要です。

次の設定により、VxFS ファイルシステムのライトバックキャッシュに SmartIO を設定できます。

- 正確に 2 つのノードを備えた SFCFSHA (Storage Foundation Cluster File System High Availability) クラスタクラスタにノードが 2 つを超える場合は、ライトバックキャッシュは有効になりません。writeback モードのキャッシュを設定しているときに別のノードを追加すると、ライトバックキャッシュは無効になります。キャッシュは読み取りモードで継続的に実行されます。

writeback モードのキャッシュを設定しているときに 2 つのノードのいずれかでクラスタファイルシステムをマウント解除すると、ライトバックキャッシュが無効になります。キャッシュは読み取りモードで継続的に実行されます。2 番目のノードでクラスタファイルシステムを再マウントすると、ライトバックキャッシュが自動的に有効になります。

- ローカルマウント構成

CFS の場合には、ライトバックキャッシュは LLT トランスポートを使ってライトバックデータをミラー化します。キャッシュされるアプリケーションの書き込みは、書き込みが戻る前にリモートキャッシュにも書き込まれます。

スループットに与える影響を避けるために 10GigE や RDMA のような広帯域幅ネットワーク経由で LLT を設定することを推奨します。

LLT の設定について詳しくは、『Symantec Storage Foundation Cluster File System High Availability インストールガイド』を参照してください。

ファイルシステムは次の特性を備える必要があります。

- ファイルシステムがディスクレイアウトバージョン 10 以降を搭載している必要があります。
- VxVM ボリュームにマウントする必要があります。



## VxFS ファイルシステムの SmartIO ライトバックキャッシュの設定

writeback モードでは、SmartIO 機能により VxFS ファイルシステムの読み込みと書き込みの I/O をキャッシュに保存します。VxFS ファイルシステムのライトバックキャッシュ用に SmartIO を設定するには、キャッシュ領域を作成して writeback モードでファイルシステムをマウントします。

### VxFS ファイルシステムの SmartIO ライトバックキャッシュの設定

- 1 次のコマンドのいずれを使って、SSD デバイスで VxFS キャッシュ領域を作成します。

- デバイスのディスクアクセス名 (*daname*) を使って 1 つ以上のデバイスを指定します。デバイスは VxVM 用に初期化し、cdsdisk 形式である必要があります。

```
# sfcache create [-t VxFS] [size] daname[...] [--auto|--noauto] ¥  
  [--nostripe|ncols=N] [cachearea_name]
```

ここで、

*daname* は、キャッシュ領域を作成するデバイスのディスクアクセス名を指定します。

*cachearea\_name* は、キャッシュ領域のカスタマイズした名前です。指定しない場合、SmartIO 機能によりキャッシュ領域の名前が自動的に生成されます。

*size* はキャッシュ領域のサイズを指定します。デフォルトでは、デバイスで利用可能な容量のすべてがキャッシュ領域に使われます。

--noauto|--auto はキャッシュ領域のタイプを指定します。デフォルトは --auto です。

次に例を示します。

```
# sfcache create ssd0_0
```

--nostripe|ncols=*n* はキャッシュ領域のレイアウトオプションを指定します。デフォルトでは、複数のディスクを指定するとキャッシュ領域はストライプボリューム上に作成されます。ncols=*n* オプションを使ってストライプボリュームの列数を指定します。指定したディスクの連結ボリュームにキャッシュ領域を作成するには --nostripe オプションを使います。

- キャッシュ領域に使うディスクグループとボリュームの名前を指定することもできます。この場合、サイズは指定できません。ボリューム全体がキャッシュ領域に使われます。

```
# sfcache create [-t VxFS] [--noauto|--auto] dg/vol
```

ここで、

--noauto|--auto はキャッシュ領域のタイプを指定します。デフォルトは --auto です。

`dg/vol1` は、キャッシュ領域に使うディスクグループ名とボリューム名を指定します。  
**SmartIO** はキャッシュ領域にこの名前を使います。  
 次に例を示します。

```
# sfcache create --auto ssd_dg/ssd_vol
```

- 2 VxFS ファイルシステムをマウントして `smartiomode` オプションを `writeback` に設定します。ファイルシステムをすでにマウント済みの場合、ファイルシステムを再マウントして、`smartiomode` オプションを `writeback` に設定する必要があります。

ローカルマウントの例:

```
# mount -t vxfs -o smartiomode=writeback /dev/vx/dsk/testdg/voll ¥  
/mnt1
```

再マウントの例:

```
# mount -t vxfs -o remount,smartiomode=writeback ¥  
/dev/vx/dsk/testdg/voll /mnt1
```

クラスタファイルシステムの場合、`smartiomode` と `cluster` オプションを設定した両方のノードでファイルシステムをマウントする必要があります。

これらのオプションを設定して **CFS** ファイルシステムをマウントすると、**SmartIO** は他のノードの **SSD** キャッシュにキャッシュを自動的に反映します。

**CFS** マウントの例:

```
# mount -t vxfs -o cluster,smartiomode=writeback ¥  
/dev/vx/dsk/testdg/voll /mnt1
```

ライトバックキャッシュを有効にするには、キャッシュ領域が `auto` または `noauto` であるかにかかわらず `smartiomode` オプションを `writeback` に設定する必要があります。キャッシュ領域が `auto` であり、`smartiomode` を設定しない場合、ファイルシステムのデフォルトの読み取りモードで **SmartIO** キャッシュが有効になります。

- 3 必要に応じて、キャッシュの動作をさらにカスタマイズすることができます。
- p.21 の「[キャッシュ動作のカスタマイズ](#)」を参照してください。
  - p.55 の「[ライトバックキャッシュのチューニング](#)」を参照してください。

## VxFS キャッシュ領域の確認およびキャッシュの監視(ライトバックモード)

**SmartIO** 機能を設定すると、キャッシュ領域が存在すること、キャッシュが発生していることを検証できます。

VxFS のキャッシュ領域では、`sfcache list` コマンドはファイルやディレクトリのキャッシュモードを示します。モードを明示的に設定しないと、ファイルやディレクトリはマウントポイントのキャッシュモードを継承します。ファイルやディレクトリにモードを明示的に設定すると、その値をすべての再マウントで継続して使います。表示されるキャッシュモードは、マウントポイントで有効なモードとは異なることがあります。writeback モードは、ファイルシステムを writeback モードでマウントしなければ有効になりません。ファイルやディレクトリを writeback モードに設定してもファイルシステムを別のモードでマウントしている場合には、ファイルやディレクトリはマウントポイントのキャッシュモードを継承します。

## キャッシュ領域の検証と監視を行うには

### 1 システムのキャッシュ領域に関する情報を表示するには

```
# sfcache list
```

たとえば、単一ノードの VxFS キャッシュ領域に次のような出力が表示されます。

```
NAME           TYPE SIZE  ASSOC-TYPE STATE  DEVICE
sfcachearea_2 VxFS 9.96g AUTO      ONLINE ssd0_0
```

writeback が有効になっているクラスタファイルシステムの場合は、writeback を有効にしてマウントした直後 (読み込みキャッシュが起きる前) に sfcache list コマンドを実行すると、キャッシュで使用済みの領域が **1.0 GB** であることが出力に示されます。キャッシュリフレクションは、それぞれのログサイズが **512 MB** のローカルログとリモートログで設定します。

### 2 特定のキャッシュ領域に関する情報を表示するには

```
# sfcache list sfcachearea_2
```

```
Cachearea: sfcachearea_2
Assoc Type: AUTO
Type: VxFS
Size: 9.96g
State: ONLINE
Layout: CONCAT
Number of Columns: 0
```

```
/dev/vx/dsk/cachedg/ssdvol:
```

FSUUID	SIZE	MODE	MOUNTPOINT
5dfe4a52805b0b003c7600002cde6d2ce70300005dfe4a52	551.0 MB	writeback	/mnt1
5efe4a52eb76000041760000a0dec33fe70300005efe4a52	39.0 MB	read	/mnt2
5efe4a524a86050046760000b485af64e70300005efe4a52	551.0 MB	writeback	/mnt3

出力には、キャッシュ構成および属性に関する情報が表示されます。

**3.47** の「[キャッシュ領域に関する情報の表示](#)」を参照してください。

### 3 特定のファイルシステムに関する情報を表示するには

```
# sfcache list /mnt1
```

```
/mnt1:
READ CACHE  WRITEBACK  MODE      PINNED  NAME
    39.0 MB      0 KB      writeback yes      /mnt1/dir
    39.0 MB      0 KB      writeback yes      /mnt1
```

4 キャッシュ使用率の統計を表示するには、次のコマンドを使います。

```
# sfcache stat sfcachearea_2

TYPE: VxFS
NAME: sfcachearea_2
      Cache Size:      9.97 GB
      Cache Utilization: 1.133 GB (11.37 %)
      File Systems Using Cache:      3
      Writeback Cache Use Limit:     8 GB
      Writeback Flush Timelag:      10 s

Read Cache                                Writeback
Hit Ratio  Data Read  Data Written  Hit Ratio  Data Written

Total:
  0.00 %      0 KB      195.0 MB  100.00 %      78.0 MB

/mnt1:
  0.00 %      0 KB      78.0 MB  100.00 %      39.0 MB

/mnt2:
  0.00 %      0 KB      39.0 MB   0.00 %           0 KB

/mnt3:
  0.00 %      0 KB      78.0 MB  100.00 %      39.0 MB
```

出力には、キャッシュに格納されたデータに対する統計が表示されます。

p.57 の「[SmartIO キャッシュ統計の表示](#)」を参照してください。

- 5 特定のファイルシステムに対しキャッシュ使用率の統計を確認するには、次のコマンドを使用します。

```
# sfcache stat /mnt1

Cache Size:      9.97 GB
Cache Utilization: 551.0 MB ( 5.40 %)

Read Cache
Hit Ratio      Data Read      Data Written      Writeback
Hit Ratio      Data Read      Data Written      Hit Ratio      Data Written

/mnt1:
0.00 %         0 KB           78.0 MB          100.00 %      39.0 MB
```

- 6 syslog を調べて writeback モードのキャッシュが有効であるかどうかを確認します。

syslog に次のような行が表示されます。

```
vxfs: msgcnt 4 writeback caching is enabled for /dev/vx/dsk/testdg/voll
```

特定のファイルシステムの writeback モードキャッシュが無効になっている場合は、syslog に次のような行が表示されます。

```
vxfs: msgcnt 9 writeback caching is disabled for /dev/vx/dsk/testdg/voll
```

## VxFS ファイルシステムの Oracle データベースの SmartIO キャッシュについて

SmartIO には、VxFS ファイルシステムで動作しているデータベースのキャッシュを最適化するアプリケーションテンプレートが用意されています。SmartIO はテンプレートを使って、データベースにある特定のタイプの情報にポリシーを適用します。たとえば、インデックスファイルにデータファイルとは異なるキャッシュポリシーがある場合があります。

SmartIO では、VxFS ファイルシステムに次のアプリケーションテンプレートを使うことができます。

- テンプレート名: **oracle**  
VxFS ファイルシステムで動作する Oracle データベースのキャッシュを最適化します。

## SmartIO plug-in for Oracle を使うための前提条件と設定

Storage Foundation and High Availability Solutions の SmartIO 機能には、Oracle データベースのプラグインが備わっています。

SmartIO plug-in for Oracle には Oracle バージョン 11 以降が必要です。この制限は、プラグインを使わない SmartIO キャッシュには適用されません。

SmartIO plug-in for Oracle を使えるようにシステムを正しく設定していることを確認します。

SmartIO plug-in for Oracle が使えるようにシステムを設定するには

- 1 Oracle SmartIO プラグインを実行する前に、ファイルシステムのキャッシュ領域を作成してキャッシュ領域をオンラインにします。
- 2 Oracle SmartIO プラグインは、データベースカタログテーブルをクエリーする必要があります。Oracle データベースがオンラインで、Oracle プラグインの `sfcache` コマンドを実行する必要があるホストと同じホストで実行していることを確認します。
- 3 `sfcache app oracle` コマンドが働くようにするには `/etc/oratab` ファイルが存在し、次の行を含む必要があります。

```
oraclesid:oracle_home:Y|N:
```

ここで、

`oraclesid` は、サーバーの Oracle インスタンスのシステム ID (SID) です。

`oracle_home` は、このインスタンスと関連付けられた `ORACLE_HOME` ディレクトリです。

`Y|N` フラグは、ブート時にインスタンスを自動的に開始するかどうかを示します。

## VxFS ファイルシステムで動作しているデータベースに対するデフォルトの SmartIO キャッシュポリシーの設定

SmartIO には、VxFS ファイルシステムで動作しているデータベースのキャッシュを最適化するアプリケーションテンプレートが用意されています。SmartIO ではテンプレートを使用して、データベースの特定のタイプのファイルにポリシーを適用します。たとえば、インデックスファイルおよびデータファイルに対するキャッシュポリシーです。

次のように、Oracle テンプレートでは Oracle データベースに対しデフォルトのポリシーが設定されます。

- ARCHLOG ファイルに対するキャッシュをオフにします (キャッシュなしモード)
- TEMPFILE に対し読み込みキャッシュ (読み取りモード) を設定します。
- OLTP データベースの場合、最も読み込まれる頻度の高いデータファイルに対して読み込みキャッシュを設定します。

OLAP データベースの場合、INDEX を含むすべてのデータファイルに対して読み込みキャッシュを設定します。



データベースに対しデフォルトの SmartIO キャッシュポリシーを設定するには

- ◆ データベースに対しデフォルトの **SmartIO** キャッシュポリシーを設定するには、次のコマンドを **ORACLE** ユーザーとして実行します。

```
# sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o setdefaults --type={OLTP | OLAP}
```

ここで、

**\$ORACLE\_HOME** と **\$ORACLE\_SID** は必須で、一意にデータベースを識別します。

**OLAP** または **OLTP** はアプリケーション負荷のタイプを示します。**OLAP (Online Analytical Processing)** アプリケーションは、多次元分析クエリを意図するワークロードを処理します。**OLTP (Online Transaction Processing)** アプリケーションは、データ入力や取得トランザクション処理などのトランザクション指向ワークロードを処理します。

**OLTP** データベースの例:

```
$ sfcache app cachearea=sfcachearea_1 oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o setdefaults --type=oltp  
  
INFO: Oracle Instance tpcc is running  
INFO: Store DB details at /tpccdata/.CACHE_INFO  
INFO: Setting oltp policies  
INFO: Setting nocache mode to /tpccdata  
  
INFO: Setting nocache mode to /tpcclog  
  
INFO: Setting nocache mode to /tpcclog/log_1_1  
INFO: Setting nocache mode to /tpcclog/log_1_2  
INFO: Setting nocache mode to /tpcclog/log_1_3  
INFO: Setting nocache mode to /tpcclog/log_1_4  
  
INFO: Setting nocache mode to /home/oracle/app/oracle/product/11.2.0  
/dbhome_1/dbs/arch  
INFO: Setting read mode to /tpccdata/temp_0_0  
INFO: Setting read mode to /tpccdata/temp_0_2  
INFO: Setting read mode to /tpccdata/temp_0_1  
INFO: Setting read mode to /tpccdata/temp_0_4  
  
AWR snapid range min=1 max=7  
  
INFO: Setting read mode to /tpccdata/stok_0_25  
INFO: Setting read mode to /tpccdata/stok_0_24  
INFO: Setting read mode to /tpccdata/stok_0_20
```

```
INFO: Setting read mode to /tpccdata/stok_0_29
INFO: Setting read mode to /tpccdata/stok_0_23
INFO: Setting read mode to /tpccdata/stok_0_22
INFO: Setting read mode to /tpccdata/cust_0_5
```

**OLAP データベースの例:**

```
$ sfcache app cachearea=sfcachearea_1 oracle -S $ORACLE_SID ¥
-H $ORACLE_HOME -o setdefaults --type=olap
INFO: Oracle Instance tpcc is running
INFO: Store DB details at /tpccdata/.CACHE_INFO
INFO: Setting olap policies
INFO: Setting nocache mode to /tpccdata

INFO: Setting nocache mode to /tpcclog

INFO: Setting nocache mode to /tpcclog/log_1_1
INFO: Setting nocache mode to /tpcclog/log_1_2
INFO: Setting nocache mode to /tpcclog/log_1_3
INFO: Setting nocache mode to /tpcclog/log_1_4

INFO: Setting nocache mode to /home/oracle/app/oracle/product/11.2.0
/dbhome_1/dbs/arch
INFO: Setting read mode to /tpccdata/temp_0_0
INFO: Setting read mode to /tpccdata/temp_0_2
INFO: Setting read mode to /tpccdata/temp_0_1
INFO: Setting read mode to /tpccdata/temp_0_4

INFO: Setting read mode to /tpccdata/icust2_0_30
INFO: Setting read mode to /tpccdata/ordr_0_32
INFO: Setting read mode to /tpccdata/iordr2_0_44
INFO: Setting read mode to /tpccdata/iordr2_0_29
INFO: Setting read mode to /tpccdata/iordr2_0_47
INFO: Setting read mode to /tpccdata/icust2_0_49
INFO: Setting read mode to /tpccdata/icust1_0_2
INFO: Setting read mode to /tpccdata/istok_0_1
INFO: Setting read mode to /tpccdata/ordr_0_33
INFO: Setting read mode to /tpccdata/ordr_0_37
INFO: Setting read mode to /tpccdata/iordr2_0_37
```

## データベースオブジェクトに対する SmartIO キャッシュポリシーの設定

データベース処理および使用状況の知識を持つデータベース管理者 (DBA) は、この情報に基づいて SmartIO キャッシュポリシーを調整することをお勧めします。名前が付いた表領域、最新パーティション、または特定のデータファイルを含む、指定されたデータベースオブジェクトに対し SmartIO キャッシュポリシーを設定することができます。指定されたデータベースをピン設定して、それを SmartIO キャッシュ領域に保持することもできます。

p.36 の「データベースオブジェクトのピン設定とピン解除」を参照してください。

指定されたデータベースオブジェクトに対しキャッシュポリシーを設定するには

- ◆ 次のコマンドを実行します。

```
# sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o set --cachemode={nocache|read} ¥  
{--datafile=name | --partition=name ¥  
| --tablespace=name | --tablecluster=name ¥  
| --filelist=name}
```

## データベースオブジェクトのピン設定とピン解除

この手順を使用して、名前が付いた表領域、表クラスタ、パーティションまたはデータファイルを含む、指定されたデータベースオブジェクトをピン設定またはピン解除します。

- pin 操作は、データがキャッシュから退去されないようにします。通常使うデータベースオブジェクトをピン設定すると、SmartIO はデータを退去しないので後でそのデータのキャッシュが再び必要になることはありません。ピン設定したデータは、削除または明示的にピン解除するまで無制限にキャッシュに保持されます。
- unpin 操作は、ピン設定した状態からデータを解除します。unpin 操作をしてもデータがすぐに退去することはありません。SmartIO では、キャッシュに領域が必要な場合はその他のデータと同様にこのデータの退去を検討します。

指定されたデータベースオブジェクトをピン設定またはピン解除するには

- ◆ 名前が付いた表領域、表クラスタ、パーティション、データファイルを含む、指定されたデータベースオブジェクトをピン設定またはピン解除するには、次のコマンドを使用します。

```
# sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o {pin | unpin} {--datafile=name ¥  
| --partition=name | --tablespace=name}
```

## データベースに対するキャッシュの有効化と無効化

SmartIO plug-in for Oracle により、データベースに対するキャッシュを有効化または無効化することができます。

この操作を使用して、バックアップまたはデータウェアハウジング ETL (抽出、変換、ロード) 操作などのデータベースジョブに対し一時的にキャッシュを無効化することができます。ジョブが完了した後、キャッシュを有効化することができます。データベースがオンラインの間、キャッシュの有効化と無効化が可能です。

データベースに対するキャッシュを有効化するには

- ◆ 次のコマンドを実行します。

```
# sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o enable
```

次に例を示します。

```
$ sfcache app cachearea=sfcachearea_1 oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o enable  
INFO: Oracle Instance tpcc is running  
INFO: Store DB details at /tpccdata/.CACHE_INFO  
INFO: Setting enable mode to /tpccdata  
  
INFO: Setting enable mode to /tpcclog
```

データベースに対しキャッシュを無効化するには

- ◆ 次のコマンドを実行します。

```
# sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o disable
```

次に例を示します。

```
$ sfcache app cachearea=sfcachearea_1 oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o disable  
INFO: Oracle Instance tpcc is running  
INFO: Store DB details at /tpccdata/.CACHE_INFO  
INFO: Setting disable mode to /tpccdata  
  
INFO: Setting disable mode to /tpcclog
```

## データベースに対するキャッシュポリシー詳細のリスト

この手順を使用して、データファイル、パーティションまたは表スペースを含む、指定されたデータベースオブジェクトに対しキャッシュポリシーをリストします。

データベースに対しキャッシュポリシーをリストするには

- ◆ 次のコマンドを実行します。

```
# sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o list (--datafile=name ¥  
| --partition=name | --tablespace=name)
```

たとえば、表スペース **stok\_0** に対しキャッシュポリシーをリストするには、次のように操作します。

```
$ sfcache app cachearea=sfcachearea_1 oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o list --tablespace=stok_0  
INFO: Oracle Instance tpcc is running  
INFO: Store DB details at /tpccdata/.CACHE_INFO  
FILENAME                MODE      PINNED      CACHE_USED  
-----                -
```

FILENAME	MODE	PINNED	CACHE_USED
/tpccdata/stok_0_0	read	no	1.05 GB
/tpccdata/stok_0_2	read	no	1.046 GB
/tpccdata/stok_0_1	read	no	1.055 GB
/tpccdata/stok_0_4	read	no	1.03 GB
/tpccdata/stok_0_3	read	no	1.043 GB
/tpccdata/stok_0_5	read	no	1.055 GB
/tpccdata/stok_0_6	read	no	1.044 GB
/tpccdata/stok_0_8	read	no	1.054 GB
/tpccdata/stok_0_7	read	no	1.048 GB
/tpccdata/stok_0_9	read	no	1.03 GB
/tpccdata/stok_0_10	read	no	1.029 GB
/tpccdata/stok_0_12	read	no	1.05 GB
/tpccdata/stok_0_11	read	no	1.045 GB

## データベースに対するキャッシュ統計のリスト

DBA では、データベースレベルでキャッシュ使用率およびヒット率を監視することができます。統計を使用して、現在のキャッシュポリシーを評価することができます。ヒット率に問題があれば、Oracle AWR レポートを参照し、ポリシーを変更することができます。

データベースに対しキャッシュ統計をリストするには

- ◆ データベースのすべての **mountpoint** に対するキャッシュ統計を表示するには、次のコマンドを使用します。

```
$ sfcache app cachearea=cachearea_name oracle -S $ORACLE_SID ¥  
-H $ORACLE_HOME -o stat
```

```
INFO: Oracle Instance tpcc is running  
INFO: Store DB details at /tpccdata/.CACHE_INFO  
VxFS SmartIO Statistics
```

```
-----  
Cache Size = 1.096 TB  
Cache Utilization = 64.3 GB ( 5.73 %)  
Mount          Hit Ratio  Cached Pinned  Read      Written  Data  
                Files    Files  Bytes   Bytes    Pinned  
-----  
/tpccdata      67.80 %   445    10      134.4 GB 179.0 GB 160 KB  
/tpcclog       38.10 %   9       0       4 KB     8 KB    0 KB
```

## VxVM ボリュームのデータベースの SmartIO キャッシュについて

SmartIO には、VxVM ボリュームで動作しているデータベースのキャッシュを最適化するアプリケーションテンプレートが用意されています。SmartIO はテンプレートを使って、データベースにある特定のタイプのボリュームにポリシーを適用します。たとえば、インデックスボリュームにデータボリュームとは異なるキャッシュポリシーがある場合があります。

SmartIO には VxVM ボリューム用に次のアプリケーションテンプレートが用意されています。

- テンプレート名: **oracle**  
VxVM ボリュームで動作する Oracle データベースのキャッシュを最適化します。
- テンプレート名: **sybase**  
VxVM ボリュームで動作する Sybase データベースのキャッシュを最適化します。

## VxVM ボリュームに対する SmartIO データベースキャッシュテンプレートの適用

SmartIO には、VxVM ボリュームで動作しているデータベースのキャッシュを最適化するアプリケーションテンプレートが用意されています。SmartIO はテンプレートを使って、

データベースにある特定のタイプのボリュームにポリシーを適用します。たとえば、インデックスボリュームおよびデータボリュームに対するキャッシュポリシーです。

### VxVM ボリュームに対し SmartIO sybase キャッシュテンプレートを適用するには

- 1 root ユーザーとしてログインします。
- 2 SYBASE 環境変数をエクスポートします。

```
# export SYBASE=/sybase
```

ここで /sybase は Sybase のホームディレクトリです。

- 3 次のコマンドを使用して、VxVM ボリュームに対し SmartIO sybase キャッシュテンプレートを適用します。

```
# sfcache app cachearea=cachearea_name sybase ¥  
{olap|oltp} db_usernamedb_server [db_name]
```

ここで、

olap または oltp はアプリケーション負荷のタイプを示します。OLAP (Online Analytical Processing) アプリケーションは、多次元分析クエリを意図するワークロードを処理します。OLTP (Online Transaction Processing) アプリケーションは、データ入力や取得トランザクション処理などのトランザクション指向ワークロードを処理します。

*db\_user\_name* は、データベースユーザー名を示します。

*db\_server* は、データベースサーバー名を示します。

*db\_name* は、データベース名を示します。データベース名は任意です。データベース名が指定されていない場合、コマンドはユーザー *db\_user\_name* の元ですべてのデータベースに対して実行されます。

次に例を示します。

```
# sfcache app cachearea=sfcachearea_1 sybase olap sa VXVMSYBS vmdb
```



- 4 プロンプトでデータベースユーザーのパスワードを入力します。
- 5 必要に応じて、キャッシュログファイルのコマンドアクティビティを表示することができます。

ログファイルは /etc/vx/log/sfcache.log です。

次に例を示します。

```
Enabling caching for sybdg/DB1DATA  
sfcache enable sybdg/DB1DATA  
Disabling caching for sybdg/DB1LOG  
Sfcache disable sybdg/DB1LOG
```

## VxVM ボリュームに SmartIO oracle キャッシュテンプレートを適用するには

- 1 root ユーザーとしてログインします。
- 2 次のコマンドを使用して、VxVM ボリュームに対して SmartIO oracle キャッシュテンプレートを適用します。

```
# sfcache app cachearea=cachearea_name oracle ¥  
{olap|oltp} oracle_user_nameORACLE_HOME ¥  
ORACLE_SID [ASM_HOME [ASM_SID]]
```

ここで、

**olap** または **oltp** はアプリケーション負荷のタイプを示します。**OLAP** (Online Analytical Processing) アプリケーションは、多次元分析クエリを意図するワークロードを処理します。**OLTP** (Online Transaction Processing) アプリケーションは、データ入力や取得トランザクション処理などのトランザクション指向ワークロードを処理します。

**oracle\_user\_name** は Oracle ユーザーのユーザー名を示します。

**ORACLE\_HOME** は Oracle のソフトウェアがインストールされるディレクトリを示します。通常、**ORACLE\_HOME** 値は Oracle ユーザーの `bash.rc` ファイルまたは `profile` ファイルに保存されます。

**ORACLE\_SID** はデータベースを一意に識別するシステム ID を示します。

**ASM\_HOME** は ASM ソフトウェアをインストールするディレクトリを示します。

**ASM\_SID** は ASM インスタンスのシステム ID を示します。デフォルトでは、値は `+ASM` です。ASM のインスタンスが複数の場合は値が異なることがあります。

次の例では Oracle データベースのアプリケーションコマンドの変化を示します。

最初の例は VxVM ボリュームで直接作成された Oracle データベースを示したものです。このシナリオでは、**ASM\_HOME** と **ASM\_SID** 値は適用できません。

RAW ボリューム上の Oracle の例:

```
# sfcache app cachearea=sfcachearea_1 oracle olap oracle /ora_base/db_home rawdb
```

次の例は VxVM ボリュームで作成された Oracle ASM データベースを示したものです。このシナリオでは、**ASM\_HOME** を指定する必要があります。必要に応じて、**ASM\_SID** を指定します。

Oracle ASM の例:

```
# sfcache app cachearea=sfcachearea_1 oracle oltp oracle /orabin/dbbase/dbhome  
¥ testdb /orabin/gridhome
```

- 3 プロンプトでデータベースユーザーのパスワードを入力します。
- 4 必要に応じて、キャッシュログファイルのコマンドアクティビティを表示することができます。

ログファイルは /etc/vx/log/sfcache.log です。

**RAW** ボリューム上の **Oracle** のログファイルの例:

```
Fri Jun  7 22:04:31 IST 2013 sfcache app cachearea=sfcachearea_1
oracle olap oracle /ora_base/db_home rawdb
Enabling caching for rawdg/rawvol02
/usr/sbin/vxprint -v -g rawdg -e 'v_name="rawvol02"'
/usr/sbin/vxprint -g rawdg -F %cache_area_type rawvol02
/usr/sbin/vxprint -g rawdg -F %iscachevol rawvol02
/usr/sbin/vxprint -g rawdg -F %caching rawvol02
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
Enabling caching for rawdg/rawvol06
/usr/sbin/vxprint -v -g rawdg -e 'v_name="rawvol06"'
/usr/sbin/vxprint -g rawdg -F %cache_area_type rawvol06
/usr/sbin/vxprint -g rawdg -F %iscachevol rawvol06
/usr/sbin/vxprint -g rawdg -F %caching rawvol06
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
Disabling caching for rawdg/rawvol01
/usr/sbin/vxprint -v -g rawdg -e 'v_name="rawvol01"'
/usr/sbin/vxprint -g rawdg -F %cache_area_type rawvol01
/usr/sbin/vxprint -g rawdg -F %iscachevol rawvol01
/usr/sbin/vxprint -g rawdg -F %caching rawvol01
```

**Oracle ASM** のログファイルの例:

```
Enabling caching for testdg/testvol
/usr/sbin/vxprint -v -g testdg -e 'v_name="testvol"'
/usr/sbin/vxprint -g testdg -F %cache_area_type testvol
/usr/sbin/vxprint -g testdg -F %iscachevol testvol
/usr/sbin/vxprint -g testdg -F %caching testvol
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
Enabling caching for testdg/testvol2
/usr/sbin/vxprint -v -g testdg -e 'v_name="testvol2"'
/usr/sbin/vxprint -g testdg -F %cache_area_type testvol2
/usr/sbin/vxprint -g testdg -F %iscachevol testvol2
/usr/sbin/vxprint -g testdg -F %caching testvol2
/usr/sbin/vxprint -o alldgs -q -v -e 'v_cachearea_vm=on'
```

# SmartIO の管理

この章では以下の項目について説明しています。

- キャッシュ領域の作成
- キャッシュ領域に関する情報の表示
- データオブジェクトのキャッシュの有効化または無効化
- キャッシュ領域へのデバイスの追加
- ボリュームからキャッシュ領域へのキャッシュの一時停止
- キャッシュ領域からのデバイスの削除
- キャッシュ領域の破棄
- VxVM キャッシュ領域の属性の設定
- VxFS キャッシュ領域に対するキャッシュモードの設定または変更
- ライトバックキャッシュ領域からのダーティデータのフラッシュ
- ライトバックキャッシュのチューニング
- SmartIO キャッシュ統計の表示

## キャッシュ領域の作成

SmartIO はキャッシュ領域の概念を導入します。キャッシュ領域は、SmartIO がキャッシュしたデータとそのデータに関するメタデータを格納するために使うストレージ領域です。I/O キャッシュで使うキャッシュ領域を作成できます。通常、キャッシュ領域には SSD デバイスまたはその他の高速デバイスを使います。キャッシュ領域は、VxFS キャッシュまたは VxVM キャッシュに使うことができます。各システムの各タイプごとに 1 つのみのキャッシュ領域を作成できます。

デバイスでキャッシュ領域を作成するには、デバイス名(ディスクアクセス名)またはデバイスのディスクグループとボリュームの名前を指定します。

デフォルトにより、キャッシュ領域には **auto** の関連付けのタイプが設定されています。同一タイプ (**VxVM** または **VxFS**) のすべてのデータオブジェクトは、暗黙的に自動キャッシュ領域に関連付けられます。関連付けのタイプが **noauto** である場合、データオブジェクトをキャッシュ領域に明示的に関連付ける必要があります。

クラスタの各ノードには個別のローカルキャッシュ領域があります。

### VxVM キャッシュ領域を作成するには

- ◆ 次のコマンドのいずれを使って、SSD デバイスで VxVM キャッシュ領域を作成します。
  - デバイスのディスクアクセス名 (**daname**) を使って 1 つ以上のデバイスを指定します。デバイスは VxVM 用に初期化し、**cdsdisk** 形式である必要があります。

```
# sfcache create -t VxVM [size] daname[...] ¥  
[cacheline_size=cacheline_size] [--auto|--noauto] [--nostripe|ncols=N]  
[cachearea_name]
```

ここで、

**daname** は、キャッシュ領域を作成するデバイスのディスクアクセス名を指定します。

**cachearea\_name** は、キャッシュ領域のカスタマイズした名前です。指定しない場合、SmartIO 機能によりキャッシュ領域の名前が自動的に生成されます。

**size** はキャッシュ領域のサイズを指定します。デフォルトでは、デバイスで利用可能な容量のすべてがキャッシュ領域に使われます。

**--noauto** | **--auto** はキャッシュ領域のタイプを指定します。デフォルトは **--auto** です。

**--nostripe|ncols=n** はキャッシュ領域のレイアウトオプションを指定します。デフォルトでは、複数のディスクを指定するとキャッシュ領域はストライプボリューム上に作成されます。**ncols=n** オプションを使ってストライプボリュームの列数を指定します。指定したディスクの連結ボリュームにキャッシュ領域を作成するには **--nostripe** オプションを使います。

**cacheline\_size** は、SmartIO がキャッシュに使う単位を指定します。アプリケーション I/O がデータにアクセスするとき、SmartIO は **cacheline** のサイズに基づきそのデータをキャッシュに移動します。通常、**cacheline\_size** を変更する必要はありません。

次に例を示します。

```
# sfcache create -t VxVM ssd0_0
```

複数のディスクを指定すると、デフォルトでキャッシュ領域が指定したディスク全体にストライプ化されます。次に例を示します。

```
# sfcache create -t VxVM ssd0_0 ssd0_1
```

- キャッシュ領域に使うディスクグループとボリュームの名前を指定することもできます。この場合、サイズは指定できません。ボリューム全体がキャッシュ領域に使われます。

```
# sfcache create -t VxVM [cacheline_size=cacheline_size] ¥  
  [--noauto|--auto] dg/vol
```

ここで、

**dg/vol** は、キャッシュ領域に使うディスクグループ名とボリューム名を指定します。**SmartIO** はキャッシュ領域にこの名前を使います。

**--noauto|--auto** は、キャッシュ領域の関連付けタイプを指定します。デフォルトは **--auto** です。

次に例を示します。

```
# sfcache create -t VxVM --auto ssd_dg/ssd_vol
```

### VxFS キャッシュ領域を作成するには

- ◆ 次のコマンドのいずれを使って、**SSD** デバイスで **VxFS** キャッシュ領域を作成します。
  - デバイスのディスクアクセス名 (**daname**) を使って 1 つ以上のデバイスを指定します。デバイスは **VxVM** 用に初期化し、**cdsdisk** 形式である必要があります。

```
# sfcache create [-t VxFS] [size] daname[...] [--auto|--noauto] ¥  
  [--nostripe|ncols=N] [cachearea_name]
```

ここで、

**daname** は、キャッシュ領域を作成するデバイスのディスクアクセス名を指定します。

**cachearea\_name** は、キャッシュ領域のカスタマイズした名前です。指定しない場合、**SmartIO** 機能によりキャッシュ領域の名前が自動的に生成されます。

**size** はキャッシュ領域のサイズを指定します。デフォルトでは、デバイスで利用可能な容量のすべてがキャッシュ領域に使われます。

**--noauto|--auto** はキャッシュ領域のタイプを指定します。デフォルトは **--auto** です。

次に例を示します。

```
# sfcache create ssd0_0
```

**--nostripe|ncols=n** はキャッシュ領域のレイアウトオプションを指定します。デフォルトでは、複数のディスクを指定するとキャッシュ領域はストライプボリューム上に作成されます。**ncols=n** オプションを使ってストライプボリュームの列数

を指定します。指定したディスクの連結ボリュームにキャッシュ領域を作成するには `--nostripe` オプションを使います。

- キャッシュ領域に使うディスクグループとボリュームの名前を指定することもできます。この場合、サイズは指定できません。ボリューム全体がキャッシュ領域に使われます。

```
# sfcache create [-t VxFS] [--noauto|--auto] dg/vol
```

ここで、

`dg/vol` は、キャッシュ領域に使うディスクグループ名とボリューム名を指定します。**SmartIO** はキャッシュ領域にこの名前を使います。

`--noauto|--auto` はキャッシュ領域のタイプを指定します。デフォルトは `--auto` です。

次に例を示します。

```
# sfcache create --auto ssd_dg/ssd_vol
```

## キャッシュ領域に関する情報の表示

SmartIO では、キャッシュに SSD デバイスのキャッシュ領域が使用されます。各キャッシュ領域には、キャッシュタイプ、サイズ、関連タイプ、キャッシュの状態、キャッシュに使用されるデバイスなどの属性が含まれています。キャッシュ領域には関連するデータオブジェクトもあります。

VxFS のキャッシュ領域では、`sfcache list` コマンドはファイルやディレクトリのキャッシュモードを示します。モードを明示的に設定しないと、ファイルやディレクトリはマウントポイントのキャッシュモードを継承します。ファイルやディレクトリにモードを明示的に設定すると、その値をすべての再マウントで継続して使います。表示されるキャッシュモードは、マウントポイントで有効なモードとは異なることがあります。writeback モードは、ファイルシステムを writeback モードでマウントしなければ有効になりません。ファイルやディレクトリを writeback モードに設定してもファイルシステムを別のモードでマウントしている場合には、ファイルやディレクトリはマウントポイントのキャッシュモードを継承します。

キャッシュ領域に関する情報を表示するには

- 1 システム上のキャッシュ領域に関する情報を表示するには、次のコマンドを使います。

```
# sfcache list

NAME          TYPE SIZE  ASSOC-TYPE STATE  DEVICE
sfcachearea_1 VxVM 9.91g AUTO    ONLINE  ssd0_2809
sfcachearea_2 VxFS 31.97g AUTO    ONLINE  ssd0_0
```

- 2 特定のキャッシュ領域に関する情報を表示するには、次のコマンドを使います。

VxVM キャッシュ領域の例

```
# sfcache list sfcachearea_1
Cachearea: sfcachearea_1
Assoc Type: AUTO
Type: VxVM
Size: 30.00g
Cacheline Size: 64.00k
Memory Size: 16.00m
State: ONLINE
Layout: CONCAT
Number of Columns: 0

ASSOCIATED DATA OBJECTS:

ASSOC DATAOBJECT NAME  CACHING-MODE  STATE  KSTATE
testdg/testvol1        read          ENABLED  ENABLED
testdg/testvol2        read          ENABLED  ENABLED
```

VxFS キャッシュ領域の例

```
# sfcache list sfcachearea_2
Cachearea: sfcachearea_2
Assoc Type: AUTO
Type: VxFS
Size: 31.97g
State: ONLINE
Layout: -
Number of Columns: 0

/dev/vx/dsk/sfcache_defaultdg/sfcachearea_2:
FSUUID          SIZE  MODE  MOUNTPOINT
3671ff51-9696-0000-872c-000056bcb408  4 KB  read  /mnt1
```



## データオブジェクトのキャッシュの有効化または無効化

SmartIO 機能を使うと、ボリュームまたはマウントポイントのキャッシュを無効化または有効化できます。

自動キャッシュを設定する場合、データオブジェクトのキャッシュを明示的に有効にする必要はありません。SmartIO 機能は、キャッシュ領域の同一タイプ (VxVM または VxFS) のすべてのデータオブジェクトをキャッシュします。デフォルトでは、データオブジェクトは auto のキャッシュ状態になっています。必要であれば、特定のデータオブジェクトのキャッシュを無効にすることができます。システムに auto キャッシュ領域がある場合、SmartIO 機能は auto 状態または enabled 状態のすべてのオブジェクトをキャッシュします。

自動キャッシュを設定しない場合、SmartIO 機能にキャッシュさせるデータオブジェクトのキャッシュを明示的に有効にする必要があります。システムに no-auto キャッシュ領域がある場合、SmartIO 機能は enabled 状態のオブジェクトのみをキャッシュします。SmartIO 機能はデフォルトのキャッシュ状態である auto のキャッシュ状態のデータオブジェクトをキャッシュしません。

p.49 の「[ファイルシステムのキャッシュの有効化または無効化](#)」を参照してください。

p.50 の「[データボリュームのキャッシュの有効化または無効化](#)」を参照してください。

## ファイルシステムのキャッシュの有効化または無効化

SmartIO 機能を使うと、特定のファイルシステムのキャッシュを無効化または有効化できます。クラスタファイルシステムの場合、クラスタ内の各ノードに対してコマンドを実行する必要があります。

ファイルシステムのキャッシュの有効と無効の切り替えは、キャッシュ領域がオフラインやオンラインになったり縮小したりすると永続的でなくなります。この場合には、ファイルシステムのキャッシュはマウントポイントに設定したキャッシュモードや `sfcache set mode` コマンドを使って設定したキャッシュモードによって異なります。キャッシュ状態を永続的にするには、`sfcache set mode` コマンドを使います。

p.54 の「[VxFS キャッシュ領域に対するキャッシュモードの設定または変更](#)」を参照してください。

ファイルシステムのキャッシュを有効にするには

- ◆ ファイルシステムのキャッシュを有効にするには、次のコマンドを使います。

```
# sfcache enable mount_point
```

ここで、

`mount_point` はファイルシステムのマウントポイントです。

### ファイルシステムのキャッシュを無効にするには

- ◆ ファイルシステムのキャッシュを無効にするには、次のコマンドを使います。-o purge オプションを使って、キャッシュから既存のキャッシュデータを消去します。デフォルトでは、データはページされません。

```
# sfcache disable [-o purge] mount_point
```

ここで、

**mount\_point** はファイルシステムのマウントポイントです。

### ファイルシステムのキャッシュデータをページするには

- ◆ ファイルシステムのキャッシュデータをページするには、次のコマンドを使います。purge オプションを使って、キャッシュから既存のキャッシュデータを消去します。

```
# sfcache purge {mount_point|fsuuid}
```

ここで、

**mount\_point** はファイルシステムのマウントポイントです。

**fsuuid** はファイルシステムの UUID を指定します。**fsuuid** を指定して、マウントされていないファイルシステムのキャッシュデータをページします。

## データボリュームのキャッシュの有効化または無効化

SmartIO 機能を使うと、特定のデータボリュームのキャッシュを無効化または有効化できます。

### 特定のデータボリュームのキャッシュを有効にするには

- ◆ 特定のデータボリュームのキャッシュを有効にするには、次のコマンドを使います。

```
# sfcache enable [--auto] dgname/volname
```

### 特定のデータボリュームのキャッシュを無効にするには

- ◆ 特定のデータボリュームのキャッシュを無効にするには、次のコマンドを使います。

```
# sfcache disable dgname/volname
```

## キャッシュ領域へのデバイスの追加

キャッシュ領域にデバイスを追加して、キャッシュ領域のサイズを拡大できます。この操作はオンラインで実行できます。

#### キャッシュ領域のサイズを変更するには

- ◆ キャッシュ領域のサイズを変更するには、次のコマンドを使います。ディスクアクセス名により新しいデバイスを指定し、そのデバイスでキャッシュ領域を拡大します。maxsize オプションは、デバイスによってサポートされる最大サイズまでキャッシュ領域を拡大します。

```
# sfcache resize [daname ...] {newsize | maxsize} cachearea_name
```

## ボリュームからキャッシュ領域へのキャッシュの一時停止

VxVM タイプのキャッシュ領域の場合には、ボリュームとキャッシュ間の関係を削除せずに、特定のボリュームに対しキャッシュ領域へのキャッシュを一時的に停止できます。キャッシュを一時停止し、後で再開することができます。キャッシュが一時停止されると、新しいデータは読み込み時にキャッシュに格納されません。書き込みがすでにキャッシュに格納されている領域で発生すると、キャッシュの内容は更新されます。

#### キャッシュを一時停止するには

- ◆ 一時停止するには、次のコマンドを使用します。

```
# sfcache set dgname/volname --pause
```

#### キャッシュを再開するには

- ◆ ボリューム対しキャッシュを再開するには、次のコマンドを使用します。

```
# sfcache set dgname/volname --resume
```

## キャッシュ領域からのデバイスの削除

キャッシュ領域が複数のデバイスにまたがる場合、キャッシュ領域からいずれかのデバイスを削除できます。

### キャッシュ領域からデバイスを削除するには

- 1 デバイスを削除したら、残りのデバイスにキャッシュのサイズが合うことを確認します。必要に応じて、キャッシュのサイズを残りのデバイスのサイズより小さいサイズに調整します。

VxFS のキャッシュ領域を縮小すると、キャッシュに保存したデータすべて(統計情報を含む)が退去されることがあります。

```
# sfcache resize newsizecachearea_name
```

たとえば、現在のキャッシュ領域 `sfcachearea_1` の最大サイズが **5 GB** であると仮定します。**2 GB** の SSD である `ssd0_0` を削除するとします。まず、キャッシュ領域を **2.5 GB** にサイズ変更します。

```
# sfcache resize 2.5g sfcachearea_1
```

- 2 キャッシュ領域からデバイスを削除するには、ディスクアクセス名でデバイスを指定してキャッシュ領域を削除します。

```
# sfcache rmdev [daname ...]
```

たとえば、`ssd0_0` という SSD は次のように削除します。

```
# sfcache rmdev ssd0_0
```

- 3 省略可能なオプションとして、`maxsize` オプションを使ってキャッシュ領域をキャッシュ領域内の残りのデバイス上に収まる最大サイズに変更できます。

```
# sfcache resize maxsize cachearea_name
```

次に例を示します。

```
# sfcache resize maxsize sfcachearea_1
```

## キャッシュ領域の破棄

キャッシュ領域を破棄して、キャッシュからすべてのデータを削除できます。キャッシュ領域を破棄する前に、キャッシュ領域をオフラインにしてキャッシュが実行されない状態にする必要があります。

キャッシュ領域を破棄するには

- 1 キャッシュ領域をオフラインにするには

```
# sfcache offline cachearea_name
```

- 2 キャッシュ領域を破棄するには、次のコマンドを使います。

```
# sfcache delete cachearea_name
```

## VxVM キャッシュ領域の属性の設定

VxVM キャッシュ領域に対し、次の属性を設定することができます。

- **memsz**。SmartIO ソリューションで VxVM キャッシュ関連メタデータの保持が要求されるシステム RAM の量。VxVM キャッシュ領域の作成時に、SmartIO によって最適値が計算されます。システムのシステム RAM が少なくなっている場合を除いて、この値を変更する必要はありません。
- **関連タイプ**。関連タイプは、`--auto` または `--noauto` です。関連タイプによって、キャッシュがシステムのボリュームに対し自動的に有効になるかどうかが決まります。  
p.12 の「[VxVM ボリュームの自動キャッシュ](#)」を参照してください。

メモリサイズを設定するには

- ◆ メモリサイズを設定するには、次のコマンドを使用します。

```
# sfcache set {dg/vol|cachearea_name} memsz=size
```

各オプションの説明

**dg/vol** は、キャッシュ領域に使用されるディスクグループとボリュームを指定します。

**cachearea\_name** は、キャッシュ領域の名前を指定します。

**size** は、コア内メタデータ用メモリに対する最大サイズを指定します。

次に例を示します。

```
# sfcache set myspecialcache memsz=128m
```

### 関連タイプを設定するには

- ◆ 関連タイプを設定するには、次のコマンドを使用します。

```
# sfcache set {--auto|--noauto} {dg/vol|cachearea_name}
```

各オプションの説明

**dg/vol** は、キャッシュ領域に使用されるディスクグループとボリュームを指定します。

**cachearea\_name** は、キャッシュ領域の名前を指定します。

次に例を示します。

```
# sfcache set --noauto mydg/myvol
```

## VxFS キャッシュ領域に対するキャッシュモードの設定または変更

VxFS キャッシュ領域の場合、キャッシュモードにより特定のマウントポイントに対して実行されるキャッシュの種類が決定します。モードは、`nocache`、`read` または `writeback` です。デフォルトのモードは `read` です。

VxVM キャッシュ領域は読み取りモードのみをサポートします。

キャッシュ領域のタイプ (VxVM または VxFS) を変更することはできません。キャッシュ領域を破棄し、必要なタイプの新しいキャッシュ領域を作成する必要があります。

VxFS マウントポイントのキャッシュモードを設定するには、`-o smartiomode` オプションを使います。`mount` コマンドで設定したキャッシュモードは、マウントポイントのオブジェクトに対して有効にできる最高レベルのキャッシュを表します。キャッシュなしモードを指定すると、マウントポイントに対して **SmartIO** キャッシュが無効になります。そのマウントポイントのデータオブジェクトに対して **SmartIO** キャッシュを有効にすることはできません。キャッシュを有効にするには、ファイルシステムを再マウントする必要があります。

同様に、マウント中に読み込みモードを指定すると、そのマウントポイントのデータオブジェクトに対して **SmartIO** `writeback` キャッシュを有効にできません。

### VxFS マウントポイントのキャッシュモードを設定するには

- ◆ VxFS ファイルシステムをマウントするときにキャッシュモードを設定するには、次のコマンドを使います。

```
# mount -t vxfs -o smartiomode=[mode] /dev/vx/dsk/testdg/vol1 /mnt1
```

ここで、

**mode** は次のいずれかです。

- writeback
- read
- nocache

VxFS マウントポイントのキャッシュモードを変更するには

- ◆ ファイルまたはディレクトリのキャッシュモードを変更するには、次のコマンドを使います。`mount_point` で設定されているモードよりも高いレベルのキャッシュにキャッシュモードを変更できません。たとえば、ファイルシステムをマウントするときに `nocache` モードを指定した場合、読み取りキャッシュを有効にできません。

```
# sfcache set [-r] mode=[nocache|read|writeback] {file|dir}
```

変更を再帰的にするには、`-r` オプションを使います。

## ライトバックキャッシュ領域からのダーティデータのフラッシュ

SmartIO では、キャッシュのダーティデータは通常処理中にディスクに自動的にフラッシュされます。ダーティデータは、ファイルシステムのマウントが解除されるか、またはファイルシステムのフリーズが必要になるその他の操作中にフラッシュされます。ダーティデータは、一定の間隔で定期的にフラッシュされます。この間隔は、チューニングパラメータを設定することによって調整できます。

p.55 の「[ライトバックキャッシュのチューニング](#)」を参照してください。

ファイルの `writeback` キャッシュを無効にすると、そのファイルのライトバックのダーティデータもフラッシュされます。

キャッシュからディスクへのダーティデータのフラッシュを手動でトリガする必要がある場合もあります。たとえば、データの整合性を確保するために、アレイレベルのスナップショットを作成する前にキャッシュをフラッシュします。

次のコマンドを使うと、ダーティデータのフラッシュを手動でトリガできます。

```
# sfcache flush [-r] {mount_point|directory|file}
```

`-r` オプションを使用して、選択を再帰的に行います。

## ライトバックキャッシュのチューニング

`writeback` キャッシュを有効にすると、ファイルが明示的に「キャッシュなし」とマーク付けされている場合やキャッシュが満杯の場合を除いてディスクから読み込むすべてのデータをキャッシュに保存します。書き込みの場合、特定の書き込みによりデータがキャッシュされます。ファイルをロードすることで、アプリケーションを高速化できます。キャッシュに

ファイルを固定すると、そのデータは退去させられません。一部のデータがすでにキャッシュされ、それに該当するディスクの部分が上書きされる場合、SmartIO は新しいデータをキャッシュデバイスに書き込んでキャッシュされたデータを最新のデータに保ちます。

データベーステンプレートを使うと、SmartIO はテンプレートのルールに従ってキャッシュを行います。

p.30 の「[VxFS ファイルシステムの Oracle データベースの SmartIO キャッシュについて](#)」を参照してください。

次のチューニングパラメータを使って、キャッシュのサイズとキャッシュでのデータの保持期間を調整できます。

「[ノードごとのダーティデータに使う最大容量の設定](#)」

「[ダーティデータに使う最大保持期間の設定](#)」

## ノードごとのダーティデータに使う最大容量の設定

ライトバックを有効にすると、ダーティデータに使うキャッシュの量を設定できます。

`writeback_size` 属性は、各ファイルシステムのライトバックデータに使うキャッシュ領域の最大容量を設定します。最大値はノードごとに設定します。デフォルトでは、最大値はありません。最大値を設定する場合は、値を少なくとも **512 MB** にする必要があります。

クラスタファイルシステムでは、`writeback` モードの SmartIO は他のノードの SSD キャッシュに各ノードのキャッシュデータを反映 (ミラー化) します。実際のディスク領域の使用率は、スタンドアロンファイルシステムに使う量の **2 倍** です。ただし、反映データは最大サイズを考慮しません。たとえば、`writeback_size` を **512 MB** に設定すると、合計が **1024 MB** でもクラスタファイルシステムは各ノードに最大 **512 MB** 使います。

次のコマンドを実行して、最大値を設定します。クラスタファイルシステムの場合、クラスタの各ノードに対してコマンドを実行して、クラスタ全体の設定を行います。

```
# sfcache set writeback_size=size
```

次に例を示します。

```
# sfcache set writeback_size=lg
```

次のコマンドを使って、現在値を表示します。

```
# sfcache stat cachearea_name
```

p.57 の「[SmartIO キャッシュ統計の表示](#)」を参照してください。

## ダーティデータに使う最大保持期間の設定

ダーティデータとは、ディスクにフラッシュされていないキャッシュ内のデータで、データディスクと同期しない状態です。保持期間により、ダーティデータがフラッシュされていない状態を継続する期間を決定します。デフォルトは **10 秒** です。



クラスタファイルシステムの場合、クラスタの各ノードに対してコマンドを実行して、クラスタ全体の設定を行います。

```
# sfcache set writeback_interval=interval
```

次に例を示します。

```
# sfcache set writeback_interval=100
```

次のコマンドを使って、現在値を表示します。

```
# sfcache stat cachearea_name
```

p.57 の「[SmartIO キャッシュ統計の表示](#)」を参照してください。

## SmartIO キャッシュ統計の表示

システムのキャッシュ統計を表示するには `sfcache stat` コマンドを使います。

表 3-1 では、出力のキャッシュ統計について記述します。

表 3-1 キャッシュ統計

フィールド	説明
HIT RATIO (VxVM キャッシュ)	キャッシュから満たされる合計 I/O の割合。読み取りと書き込みに対して表示されます。
ART(Hit)ms (VxVM キャッシュ)	キャッシュから満たされた I/O の平均応答時間。読み取り (RD) と書き込み (WR) に対して表示されます。
ART(Miss)ms (VxVM キャッシュ)	キャッシュによって満たされない I/O の平均応答時間。読み取り (RD) と書き込み (WR) に対して表示されます。
BYTES (VxVM キャッシュ)	読み取り (RD) と書き込み (WR) の合計 I/O サイズ。
NAME (VxVM キャッシュ)	キャッシュ領域の名前。
TYPE (VxVM キャッシュ)	キャッシュ領域が VxVM または VxFS であるかを示します。
%CACHE (VxVM キャッシュ)	データに現在使われているキャッシュ領域の割合。すべてのデータオブジェクトを対象とします。
Cache Size (VxFS キャッシュ)	キャッシュ領域のサイズ。

フィールド	説明
Cache Utilization (VxFS キャッシュ)	データに現在使われているキャッシュ領域の割合。
File Systems Using Cache (VxFS キャッシュ)	キャッシュを使うファイルシステムの数。
Writeback Cache Use Limit (VxFS キャッシュ)	ライトバックに使われるキャッシュ領域のサイズ。 writeback_size 属性でサイズを設定できます。サイズを設定しないとフィールドは無制限に表示されます。 p.55 の「ライトバックキャッシュのチューニング」を参照してください。
Writeback Flush Timelag	データがキャッシュに書き込まれるタイミングとディスクにフラッシュされるタイミングの間隔。 Writeback Flush Timelag の値が小さい(10 秒など)場合、sfcache の統計は表示されません。ディスクへのデータのフラッシュの方がより高速です。この場合、WB Hit Ratio に基づきキャッシュ使用率を判別できます。
Hit Ratio (VxFS キャッシュ)	キャッシュから満たされる合計 I/O の割合。読み取りと書き込みに対して表示されます。
Data Read (VxFS キャッシュ)	キャッシュからの読み取られたデータ。
Data Written (VxFS キャッシュ)	キャッシュに書き込まれたデータ。
Files Cached (VxFS キャッシュ)	キャッシュに存在するファイルの数。
Files Pinned (VxFS キャッシュ)	キャッシュに固定されたファイルの数。
Data Pinned (VxFS キャッシュ)	キャッシュに固定されたデータの量。

キャッシュ統計を表示するには

- ◆ 次のコマンドを実行します。

```
# sfcache stat
```

```

                                HIT RATIO   ART(Hit)ms   ART(Miss)ms   BYTES
NAME          %CACHE RD    WR    RD    WR    RD    WR    RD    WR
TYPE: VxVM
sfcachearea_1  13.43  91.24  94.20  0.142 0.819 0.414 0.798 15.31g 4.21g

```

TYPE: VxFS

NAME: sfcachearea\_2

Cache Size: 48.0 GB

Cache Utilization: 72.2 MB ( 0.15 %)

File Systems Using Cache: 1

Writeback Cache Use Limit: Unlimited

Writeback Flush Timelag: 10 s

Read Cache

```

Hit Ratio  Data Read  Data Written  Files Cached  Files Pinned  Data Pinned
0.00 %    0 KB        0 KB         0             0             0 KB

```

## VxVM キャッシュエリアの詳細なキャッシュ統計の表示

VxVM キャッシュエリアの詳細なキャッシュ統計を表示するには

- ◆ 次のコマンドを実行します。

```
# sfcache stat sfcachearea_1
```

```

                                HIT RATIO   ART(Hit)ms   ART(Miss)ms   BYTES
NAME          %CACHE RD    WR    RD    WR    RD    WR    RD    WR
TYPE: VxVM
sfcachearea_1  13.43  91.24  94.20  0.142 0.819 0.414 0.798 15.31g 4.21g

```

ASSOCIATED DATA OBJECTS:

```

testdg/testvol1  6.10  90.00  96.00  0.141 0.459 0.348 0.448 6.77g 1.89g
testdg/testvol2  7.32  91.00  92.00  0.143 1.179 0.480 1.149 8.54g 2.31g

```

## VxFS キャッシュ領域に対する詳細キャッシュ統計の表示

VxFS キャッシュ領域の場合には、ファイルシステムをマウント解除したりマウントした後で統計は変更されません。クラスタファイルシステムに対し、統計はクラスタノードを再ブートした後は変わりません。

**VxFS キャッシュ領域に対し詳細キャッシュ統計を表示するには**

- 1 次のコマンドを実行します。

```
# sfcache stat sfcachearea_2
```

```
TYPE: VxFS
```

```
NAME: sfcachearea_2
```

```
Cache Size: 9.97 GB
```

```
Cache Utilization: 1.133 GB (11.37 %)
```

```
File Systems Using Cache: 3
```

```
Writeback Cache Use Limit: 8 GB
```

```
Writeback Flush Timelag: 10 s
```

Read Cache			Writeback	
Hit Ratio	Data Read	Data Written	Hit Ratio	Data Written
Total:				
0.00 %	0 KB	195.0 MB	100.00 %	78.0 MB
/mnt1:				
0.00 %	0 KB	78.0 MB	100.00 %	39.0 MB
/mnt2:				
0.00 %	0 KB	39.0 MB	0.00 %	0 KB
/mnt3:				
0.00 %	0 KB	78.0 MB	100.00 %	39.0 MB

2 詳細を参照するには、次のように -l オプションを使います。

```
# sfcache stat -l
TYPE: VxFS
NAME: cachedg_lpr1/cachevol
      Cache Size:      200 GB
      Cache Utilization: 714.0 MB ( 0.35 %)
File Systems Using Cache:      5
Writeback Cache Use Limit: Unlimited
      Writeback Flush Timelag: 300 s

Read Cache
Hit Ratio Data Read Data Written Files Cached Files Pinned Data Pinned Hit Ratio Data Written
Total:
  43.42 %   297 MB   116 MB           6           3           5 MB   34.24 %   200.5 MB

/mnt1:
  43.42 %   297 MB   116 MB           6           3           5 MB   34.24 %   200.5 MB
```

# トラブルシューティングとエラー処理

この章では以下の項目について説明しています。

- 永続または「ウォーム」VxVM キャッシュのサポート
- キャッシュ領域がディスク障害後に失われる (3158482)
- 再ブート後にキャッシュがオンラインにならない
- ノードエラー後のライトバックキャッシュのリカバリ

## 永続または「ウォーム」VxVM キャッシュのサポート

ウォームキャッシュとは、キャッシュの内容が計画された再ブート後も永続的に保持されることを意味します。デフォルトで、SmartIO では VxVM キャッシュに対しウォームキャッシュの機能は提供されません。キャッシュ領域メタデータはシステム停止時にフラッシュされません。キャッシュはシステム再ブート後、およびボリュームの再起動が必要になるたびに無効になります。この動作は「コールド」キャッシュといえます。

データの不整合が発生することがあるため、ウォームキャッシュの設定は推奨していません。ただしある状況では、適切に行えばウォームキャッシュの設定は有益である場合があります。たとえば、制御され、計画された再ブートの場合には、ウォームキャッシュを明示的に有効にすることがあります。この機能により、メタデータをキャッシュにフラッシュすることができます。これによりウォームキャッシュが作成されます。

永続キャッシュやウォームキャッシュを有効にすると、キャッシュがオフラインのときにデータボリュームを更新した場合には VxVM は永続キャッシュを検出して無効にします。クラスタ化環境で、ノードのキャッシュ領域をそのノードの計画シャットダウン時に維持して別のノードでデータボリュームを更新すると、ウォームキャッシュの内容を古いと見なして維持したキャッシュデータが退去されます。

計画された再ブートに対しウォームキャッシュを有効にするには

- 1 システム再ブートの前に、すべてのアプリケーションを終了します。
- 2 次のコマンドを実行して、ウォームキャッシュを作成します。

```
# sfcache offline --flushmeta cachearea_name
```

必要に応じて、このコマンドをシャットダウンスクリプトに追加できます。

アプリケーション再起動またはシステム再ブートの前に上記のコマンドを実行する必要があります。実行しない場合、キャッシュのデータがバージされ、キャッシュは再ブート後に作成されません。

## 古いキャッシュを持つプライマリボリュームのエラーによるデータ破損の可能性

プライマリボリュームのデータを復元する必要がある場合、ウォームキャッシュの存在により古いキャッシュが使用されることがあります。場合によっては、古いキャッシュによりデータ破損が発生する可能性があります。この状況は一般的ではありませんが、次のような復元方法が使用される場合は発生する可能性があります。

- アレイスナップショットからの復元
- NetBackup または他のバックアップソフトウェアを使用した、バックアップからプライマリ LUN への復元。
- VxVM 設定のバックアップと復元。

## HA フェールオーバー時のキャッシュの移行はサポートされない

ウォームキャッシュが有効であっても、SmartIO はフェールオーバー時に高可用性のクラスタのキャッシュを移行するための機能を提供しません。フェールオーバー後、アプリケーションがウォームキャッシュを利用することはできません。

## キャッシュ領域がディスク障害後に失われる (3158482)

SmartIO は、1 つの VxFS キャッシュ領域と 1 つの VxVM キャッシュ領域をサポートします。1 つのキャッシュ領域を作成し、ディスクにエラーが発生すると、キャッシュ領域は無効になります。キャッシュディスクグループが有効になる前にその他のタイプの 2 つ目のキャッシュ領域を作成しようとすると、最初のキャッシュ領域は失われます。オンラインにすることはできません。

たとえば、最初に VxFS キャッシュ領域を作成したとします。ディスクにエラーが発生し、そのキャッシュ領域が無効になります。ここで VxVM キャッシュ領域を作成します。VxVM キャッシュ領域の作成時に、SmartIO は既存のデフォルトのキャッシュ領域を検索します。エラーが発生したディスクのため、既存のキャッシュ領域は見つかりません。そのた



め、SmartIO は同じ名前の VxVM キャッシュ領域を作成します。ここで、VxFS キャッシュ領域を含むディスクが起動されても、SmartIO は元のキャッシュ領域にアクセスすることができません。このシナリオでは、VxFS キャッシュ領域は失われます。この場合はキャッシュ領域が失われても、データ損失またはデータ不整合の問題は発生しません。

回避策:

新しい VxFS キャッシュ領域を作成します。

## 再ブート後にキャッシュがオンラインにならない

通常、SmartIO キャッシュは、システムの再ブート後に自動的にオンラインになります。

再ブート後に SSD ドライバモジュールが自動的にロードされない場合は、ドライバをロードしてキャッシュディスクグループを手動でオンラインにする必要があります。

再ブート後にキャッシュをオンラインにするには

- 1 `insmod` コマンドを使って、SSD ドライバモジュールを読み込みます。

詳しくは、Linux 社のマニュアルを参照してください。

- 2 OS デバイスのスキャンを実行します。

```
# vxdisk scandisks
```

- 3 キャッシュを手動でオンラインにします。

```
# vxdg import cachedg
```

## ノードエラー後のライトバックキャッシュのリカバリ

writeback モードでは、SmartIO はデータをディスクに書き込む前にキャッシュに保存します。キャッシュされたデータを含むデバイスが失敗した場合には、システムがオンラインに復帰したときにキャッシュ領域に残っているデータをディスクにフラッシュする必要があります。キャッシュされたデータをフラッシュするまで、ライトバックモードでキャッシュされるファイルデータの一部分がディスクに存在しないことがあります。

ライトバックのダーティデータを含む SmartIO キャッシュは、フラッシュに利用できない場合もあります。たとえば、キャッシュ領域を含む SSD デバイスでエラーが発生する場合もあります。キャッシュデバイスがオンラインに復帰すると、そのようなファイルのダーティデータはフラッシュ処理からスキップされます。保留中のダーティデータを含むファイルにはアクセスできません。これらのファイルに対する削除を除くすべての I/O 操作は I/O エラー (EIO) を返します。

Syslog に次のエラーメッセージが表示されます。

```
Writeback cache recovery is failed for mounted_device with error  
error_code:
```

このようなファイルへのアクセスを復元する必要がある場合は、次の手順を実行します。

### アクセスできないファイルへのアクセスの復元

- 1 ファイルへのアクセスを復元するには、次のコマンドを使います。このコマンドにはルート権限が必要です。

```
# sfcache restore-access -r {mount_point|directory|file}
```

次に例を示します。

```
# /usr/sbin/sfcache restore-access /testFS
```

クラスタファイルのシステムでは、クラスタの各ノードで `sfcache restore-access` コマンドを実行します。

- 2 ファイルやディレクトリへのアクセスをリストアした後、マウントポイントのアクセスをリストアします。

```
# sfcache restore-access -r mount_point
```

次に例を示します。

```
# /usr/sbin/sfcache restore-access /testFS
```

- 3 ファイルシステムに対するライトバックキャッシュを有効にするには、次のコマンドを使います。

```
# sfcache disable /testFS
```

```
# sfcache enable /testFS
```

# コマンドリファレンス

この付録では以下の項目について説明しています。

- [SmartIO コマンドリファレンス](#)

## SmartIO コマンドリファレンス

表 A-1 では、SmartIO 機能を使うためのコマンドのリストを示します。

[sfcache\(1M\) マニュアルページ](#)を参照してください。

表 A-1 SmartIO コマンドリファレンス

コマンド	説明
<code>sfcache app</code>	指定したテンプレート名を適用します。
<code>sfcache create</code>	キャッシュ領域を作成します。
<code>sfcache delete</code>	指定したキャッシュ領域を削除します。
<code>sfcache disable</code>	指定したデータオブジェクトに対してキャッシュを無効にします。
<code>sfcache enable</code>	指定したデータオブジェクトに対してキャッシュを有効にします。
<code>sfcache flush</code>	このファイルシステムまたはキャッシュのライトバックデータをフラッシュします。
<code>sfcache list</code>	キャッシュされるファイルシステムまたはボリュームとそのキャッシュ使用率を表示します。
<code>sfcache load</code>	指定したファイルをキャッシュ領域にロードします。

コマンド	説明
<code>sfcache maxsize</code>	キャッシュ用にすでにプロビジョニングされているデバイスの空き容量を表示します。
<code>sfcache offline</code>	VxFS または VxVM によるキャッシュ領域の使用を停止します。
<code>sfcache online</code>	キャッシュ領域が利用可能であることを明示的に示します。
<code>sfcache pin</code>	ファイルまたはディレクトリが削除、切り捨て、固定解除されるまでそのファイルまたはディレクトリを保留することをマークします。
<code>sfcache purge</code>	指定したファイルシステムのキャッシュされた内容を削除します。
<code>sfcache resize</code>	指定したキャッシュ領域のサイズを変更します。
<code>sfcache restore-access</code>	ライトバックデータが欠けているファイルへの読み取りまたは書き込みアクセスを有効にします。このコマンドにより、欠けているデータが復元されることはありません。
<code>sfcache rmdev</code>	キャッシュの使用からデバイスまたは複数のデバイスを削除します。
<code>sfcache set</code>	指定した属性の値を設定します。
<code>sfcache stat</code>	キャッシュヒット率、ミス、平均読み取り/書き込みレイテンシを含むキャッシュの統計を表示します。
<code>sfcache unpin</code>	固定された状態からファイルまたはディレクトリを削除します。

## N

nocache キャッシュモード 54

## S

sfcache stat コマンド 57

SmartIO

概要 6

SSD デバイスでの VxVM のキャッシュ 13

## V

Veritas File System (VxFS)

読み取りキャッシュ 16

Veritas Volume Manager (VxVM)

自動キャッシュ 12

VxFS (Veritas File System)

キャッシュのカスタマイズ 21

読み取りキャッシュの構成 17~18

ライトバックキャッシュ 23~25

VxVM (Veritas Volume Manager)

SSD でのキャッシュ 13

読み取りキャッシュ 10

## か

書き込みキャッシュ 25

書き込みモードでの VxFS のキャッシュ 25

キャッシュからのファイルのピン解除 21

キャッシュ使用率の統計

表示 57

キャッシュ属性の設定 53

キャッシュディスクグループ 10

キャッシュ動作

カスタマイズ 21

キャッシュの一時停止 51

キャッシュの管理

キャッシュ領域の破棄 52

キャッシュの再開 51

キャッシュへのファイルのピン設定 21

キャッシュへのファイルのロード 21

キャッシュモード

設定 54

キャッシュモードの設定 54

キャッシュモードの変更 54

キャッシュ領域

監視 14, 20, 28

検証 14, 20, 28

作成 44

属性の設定 53

デバイスの削除 51

デバイスの追加 50

キャッシュ領域の拡張 50

キャッシュ領域のサイズ変更 50~51

キャッシュ領域の削除 52

キャッシュ領域の破棄 52

キャッシュ領域を減らす 51

## さ

自動キャッシュ 12

詳細キャッシュ統計の表示 59~60

ソリッドステートドライブ (SSD)

概要 6

## た

データ'83'7bリユーム

キャッシュの無効化 50

データオブジェクト 49

キャッシュの無効化 49

キャッシュの有効化 49

データボリューム

キャッシュの有効化 50

デバイスの削除 51

デバイスの追加 50

デフォルトのキャッシュ統計の表示 57

統計

キャッシュ使用率の表示 57

## は

ファイルシステム

キャッシュの無効化 49

キャッシュの有効化 49

ファイルシステムキャッシュ統計の表示 60

ボリュームキャッシュ統計の表示 59

## や

読み込みモードでの VxFS のキャッシュ 18

読み取りキャッシュ 16

読み取りキャッシュモード 54

## ら

ライトバックキャッシュ 23

ライトバックキャッシュモード 54