

Veritas InfoScale™ 7.2 SmartIO for Solid-State Drives 解决方案指南 - Linux

Veritas InfoScale SmartIO for Solid-State Drives 解决方案指南

上次更新时间： 2016-11-29

文档版本： 7.2 Rev 0

法律声明

Copyright © 2016 Veritas Technologies LLC. All rights reserved. 保留所有权利。

Veritas、Veritas 徽标、Veritas InfoScale 和 NetBackup 是 Veritas Technologies LLC 或其附属机构在美国和其他国家/地区的商标或注册商标。其他名称可能为其各自所有者的商标，特此声明。

本产品可能包含 Veritas 必需向第三方支付许可费的第三方软件（“第三方程序”）。部分第三方程序是以开放源或免费软件许可方式获得的。本软件随附的许可证协议并未改变这些开放源或免费软件许可所规定的任何权利或义务。请参考此 Veritas 产品随附的或位于以下地址的第三方法律声明：

<https://www.veritas.com/about/legal/license-agreements>

本文档中介绍的产品根据限制其使用、复制、分发和反编译/逆向工程的授权许可协议分发。未经 Veritas Technologies LLC 及其特许人（如果存在）事先书面授权，不得以任何方式任何形式复制本文档的任何部分。

本文档按“现状”提供，对于所有明示或暗示的条款、陈述和保证，包括任何适用性、针对特定用途的适用性或无侵害知识产权的暗示保证，均不提供任何担保，除非此类免责声明的范围在法律上视为无效。VERITAS TECHNOLOGIES LLC 不对任何与提供、执行或使用本文档相关的伴随或后果性损害负责。本文档所含信息如有更改，恕不另行通知。

根据 FAR 12.212 定义，授权许可的软件和文档被视为“商业计算机软件”，受 FAR Section 52.227-19 “Commercial Computer Software - Restricted Rights”（商业计算机软件受限权利）和 DFARS 227.7202 “Commercial Computer Software and Commercial Computer Software Documentation”（商业计算机软件和商业计算机软件文档）中的适用规定以及所有后续法规中规定的权利的制约，无论 Veritas 以本地服务还是托管服务提供都是如此。美国政府仅可根据本协议的条款对授权许可的软件和文档进行使用、修改、发布复制、执行、显示或披露。

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

<http://www.veritas.com>

技术支持

技术支持具有全球性支持中心。所有支持服务都将根据您的支持协议和当时有效的企业技术支持策略来提供。有关我们的支持服务以及如何联系技术支持的信息，请访问我们的网站：

<https://www.veritas.com/support>

从以下 URL 您可以管理 Veritas 帐户信息：

<https://my.veritas.com>

如果您对现有支持协议有疑问，请通过以下方式联系您所在地区的支持协议管理部门：

全球（日本除外）

CustomerCare@veritas.com

日本

CustomerCare_Japan@veritas.com

文档

请确保您具有文档的最新版本。每个文档的第 2 页显示了上次更新日期。每个指南的第 2 页提供了文档版本信息。可在 Veritas 网站上找到最新的文档：

<https://sort.veritas.com/documents>

文档反馈

您的反馈对我们很重要。请对我们的文档提出改进意见、报告错误或遗漏。请在您的报告中包括所报告的文本内容的文档标题和文档版本以及章节标题。请将反馈发送到：

doc.feedback@veritas.com

您也可以在 Veritas 社区网站上查看文档信息或提出问题：

<http://www.veritas.com/community/>

Veritas Services and Operations Readiness Tools (SORT)

Veritas Services and Operations Readiness Tools (SORT) 是一个网站，提供的信息和统计可自动处理和简化某些耗时的管理任务。根据您的产品，SORT 会帮助您准备安装和升级、识别您数据中心的风险并提高操作效率。要了解 SORT 为您的产品提供了哪些服务和工具，请参见数据表：

https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

目录

第 1 章	SFHA Solutions SmartIO 简介	6
	关于适用于固态驱动器的 SmartIO	6
	关于 SFHA 环境中的 SmartIO	7
	关于主动/主动群集环境中的 SmartIO	7
	关于 Linux 虚拟化环境中的 SmartIO	8
	关于 SmartIO 缓存探查器工具	9
第 2 章	使用 SmartIO 功能：用例	11
	关于运行在 VxVM 卷上的应用程序的 SmartIO 读取缓存	11
	为 VxVM 卷使用 SmartIO 读取缓存所需的配置	12
	针对 VxVM 卷的自动缓存	13
	为 VxVM 卷设置 SmartIO 读缓存	13
	验证 VxVM 缓存区和监视缓存	14
	关于对 VxFS 文件系统中运行的应用程序执行 SmartIO 读取缓存	17
	为 VxFS 文件系统使用 SmartIO 读取缓存所需的配置	18
	VxFS 文件系统自动缓存	18
	为 VxFS 文件系统设置 SmartIO 读缓存	18
	验证 VxFS 缓存区域并监视缓存	20
	自定义缓存行为	22
	关于使用 FSS 导出的 SSD 设备进行 SmartIO 缓存	24
	节点退出或加入群集时缓存区域的状态	25
	使用 FSS 导出的 SSD 设置缓存区域	26
	关于对 VxFS 文件系统中运行的应用程序执行 SmartIO 写回缓存	26
	为 VxFS 文件系统使用 SmartIO 写回缓存所需的配置	27
	为 VxFS 文件系统设置 SmartIO 写回缓存	28
	验证 VxFS 缓存区域并监视缓存（写回模式）	29
	关于在 VxFS 文件系统中执行读取和写回缓存的多个 SmartIO 缓存区 域	32
	关于 smartiocache 选项	36
	将 VxFS 缓存区域从一种类型转换为另一个类型	36
	在系统上设置多个缓存区域	37
	验证 VxFS 缓存区域	38
	关于 VxFS 文件系统上的 Oracle 数据库的 SmartIO 缓存	41
	对 Oracle 使用 SmartIO 插件的先决条件和配置	41

	为 VxFS 文件系统上运行的数据库设置默认的 SmartIO 缓存策略	42
	设置数据库对象的 SmartIO 缓存策略	44
	固定和取消固定数据库对象	45
	对数据库启用和禁用缓存	45
	列出数据库缓存策略详细信息	46
	列出数据库缓存统计数据	47
	关于 VxVM 卷上的数据库的 SmartIO 缓存	48
	对 VxVM 卷应用 SmartIO 数据库缓存模板	48
	技术预览: Veritas InfoScale 存储环境中的分布式 SmartIO	52
第 3 章	管理 SmartIO	58
	创建缓存区域	58
	显示有关缓存区域的信息	61
	启用或禁用对数据对象的缓存	64
	启用或禁用对文件系统的缓存	64
	启用或禁用对数据卷的缓存	65
	将设备添加到缓存区域	65
	暂停从卷缓存到缓存区域	66
	从缓存区域中删除设备	66
	销毁缓存区	67
	设置 VxVM 缓存区域的属性	68
	设置或更改 VxFS 缓存区域的缓存模式	68
	刷新写回缓存区域中的脏数据	69
	调整写回缓存	70
	查看 SmartIO 缓存统计数据	71
	查看 VxVM 缓存区的详细缓存统计数据	73
	查看 VxFS 缓存区域的详细缓存统计数据	74
第 4 章	故障排除和错误处理	77
	对持续或暖 VxVM 缓存的支持	77
	过时缓存的主卷故障可能导致数据损坏	78
	不支持在 HA 故障转移期间迁移缓存	78
	磁盘发生故障后缓存区域丢失 (3158482)	78
	重新启动后缓存未联机	79
	在节点出现故障后恢复写回缓存	79
附录 A	命令参考资料	81
	SmartIO 命令参考	81
索引	83

SFHA Solutions SmartIO 简介

本章节包括下列主题：

- [关于适用于固态驱动器的 SmartIO](#)
- [关于 SFHA 环境中的 SmartIO](#)
- [关于主动/主动群集环境中的 SmartIO](#)
- [关于 Linux 虚拟化环境中的 SmartIO](#)
- [关于 SmartIO 缓存探查器工具](#)

关于适用于固态驱动器的 SmartIO

固态驱动器(SSD)是不含旋转磁盘的设备。相比传统的旋转磁盘，当前的固态技术（例如 DRAM 和 NAND 闪存）提供了更快的数据访问速度，效率更高，而且占用的空间更小。数据中心通过多种形式运用固态技术，包括在服务器内使用，使用全闪存阵列和全闪存设备，以及将其与传统的 HDD 阵列混合使用。每种形式的价值主张各不相同。SSD 还具有许多连接类型：PCIe、FC、SATA 和 SAS。

鉴于 SSD 设备当前的每 GB 成本，SSD 的最佳值不如高容量的存储设备。采用 SSD 的优点在于能够提高性能和降低每秒每 I/O 的成本 (IOPS)。为了最大限度提高数据中心固态技术投资的回报，数据效率和放置至关重要。

Storage Foundation and High Availability Solutions (SFHA Solutions) 的 SmartIO 功能可通过 I/O 缓存提高 SSD 上的数据效率。使用 SmartIO 提高效率时，可以优化每 IOPS 的成本。SmartIO 不要求深入了解底层硬件技术。SmartIO 使用先进的可自定义启发法来确定要缓存的数据，以及从缓存删除该数据的方式。该启发法利用了 SFHA Solutions 在工作负载特性方面的知识。

SmartIO 在一个或多个目标设备上使用缓存区。缓存区是 SmartIO 用来存储缓存数据及其相关元数据的存储空间。缓存区的类型决定了缓存区支持 VxFS 缓存还是 VxVM 缓存。要开始使用 SmartIO，您可以在应用程序处于联机状态时通过单个命令创建缓存区。

当应用程序发出 I/O 请求时，SmartIO 会检查缓存能否满足该 I/O 的要求。随着应用程序从基础卷或文件系统访问数据，SmartIO 会根据内部启发法将某些数据移到缓存。后续的 I/O 会通过缓存进行处理。

对于在 VxVM 卷上装入的 VxFS 文件系统，SmartIO 支持读取和写入缓存，并且支持多种缓存模式和配置。对于在 VxVM 卷上运行的应用程序，SmartIO 还支持块级别读取缓存。

请参见第 11 页的“关于运行在 VxVM 卷上的应用程序的 SmartIO 读取缓存”。

请参见第 17 页的“关于对 VxFS 文件系统上运行的应用程序执行 SmartIO 读取缓存”。

请参见第 26 页的“关于对 VxFS 文件系统上运行的应用程序执行 SmartIO 写回缓存”。

请参见第 41 页的“关于 VxFS 文件系统上的 Oracle 数据库的 SmartIO 缓存”。

请参见第 48 页的“关于 VxVM 卷上的数据库的 SmartIO 缓存”。

关于 SFHA 环境中的 SmartIO

在群集环境中，SmartIO 缓存是群集中每个节点的本地缓存。缓存区域不会在某一节点上脱机然后在其他节点上联机。

SmartIO 读取缓存在装有 SFHA 等的活动/被动环境使用时，您可以逐出数据卷和文件系统并将其导入其他节点。SmartIO 缓存不会移至其他节点。缓存从具有本地 SmartIO 缓存区域的新节点上开始。SmartIO 不会提供在故障转移期间迁移高可用性群集中缓存的功能。

对于 VxFS 写回缓存，不建议故障转移磁盘组。在本地装入的情况下，如果没有缓存镜像，在此节点发生故障时，这个文件的磁盘副本可能是不完整或失效的。Veritas 建议改用 SFCFSHA，它可以提供缓存的数据反射。

关于主动/主动群集环境中的 SmartIO

在主动/主动环境（例如 SF Oracle RAC）中使用 SmartIO 读取缓存时，将在所有具有本地 SmartIO 缓存区域的节点上开始缓存。缓存区域可以在每个节点上以独占方式联机或脱机。

SF Oracle RAC 当前不支持 SmartIO VxFS 写回缓存。

关于 Linux 虚拟化环境中的 SmartIO

在 Linux 虚拟化环境中，当在来宾中安装 Veritas InfoScale Solutions 时，可以使用 SmartIO 将数据缓存到 SSD 或其他任何受支持的速度较快的设备。

在 KVM 和 RHEV 环境中，SmartIO 缓存不支持来宾的活动迁移。

对于 VMware，如果在 ESXi 虚拟机管理程序中启用了 DMP for VMware (SmartPool)，则 SmartIO 不支持 vMotion。

Linux 虚拟化环境中不支持 Storage Foundation for Oracle RAC。

下表显示在 Linux 虚拟化环境中如何使用 SmartIO。

表 1-1 显示在 KVM 环境中如何使用 SmartIO。

表 1-1 Linux: KVM 中对 SmartIO 的支持

来宾的配置:	主机的配置:	缓存发生:	VxVM 读取缓存	VxFS 读取缓存	VxFS 写回缓存
SF	任意 (SF 或 SFCFSHA)	在来宾中	是	是	是
SFHA	任意 (SF 或 SFCFSHA)	在来宾中	是	是	是
SFCFSHA	任意 (SF 或 SFCFSHA)	在来宾中	是	是	是
全部	SF	在主机中	是	是	是
全部	SFCFSHA	在主机中	是	是	是

表 1-2 显示在 RHEV 环境中如何使用 SmartIO。

表 1-2 Linux: RHEV 中对 SmartIO 的支持

来宾的配置:	主机的配置:	缓存发生:	VxVM 读取缓存	VxFS 读取缓存	VxFS 写回缓存
SF	任意 (SF 或 SFCFSHA)	在来宾中	是	是	是
SFHA	任意 (SF 或 SFCFSHA)	在来宾中	是	是	是
SFCFSHA	任意 (SF 或 SFCFSHA)	在来宾中	是	是	是
全部	SF	在主机中	是	是	是

来宾的配置:	主机的配置:	缓存发生:	VxVM 读取缓存	VxFS 读取缓存	VxFS 写回缓存
全部	SFCFSHA	在主机中	是	是	是

表 1-3 显示在 VMware 环境中如何使用 SmartIO。

表 1-3 Linux: VMware 中对 SmartIO 的支持

来宾的配置:	主机的配置:	缓存发生:	VxVM 读取缓存	VxFS 读取缓存	VxFS 写回缓存
SF	DMP for VMware (可选)	在来宾中	是	是	否
SFHA	DMP for VMware (可选)	在来宾中	是	是	否
SFCFSHA	DMP for VMware (可选)	在来宾中	是	是	否

有关在 Linux 虚拟化环境中配置 Veritas InfoScale Solutions 的详细信息，请参见《Veritas InfoScale™ Solutions 虚拟化指南 (Linux 版)》。

关于 SmartIO 缓存探查器工具

Smartassist 工具将对系统中的指定目标 I/O 执行指定时间的分析，计算工作量的最佳缓存大小。要从 Veritas InfoScale Operations Manager (VIOM) 运行工具，请参考 VIOM 管理指南。

Smartassist 工具支持以下目标：

- 设备路径列表
- VxVM 或 LVM 卷路径列表
- VxVM 或 LVM 卷路径列表
- 文件系统装入点列表
- Oracle/Sybase/DB2 数据库实例
- VxVM 共享卷/磁盘组列表
- CFS 装入点列表

该工具的工作过程分为两个阶段。

启动阶段	该工具将对指定目标 I/O 进行指定时间的跟踪并将跟踪输出存储到指定的目录。如果使用包含多个路径的磁盘设备，建议指定所有路径以获取正确结果。在这种情况下，使用 VxVM 或 LVM 卷设备更方便。默认情况下，启动阶段运行 3600 秒。
分析阶段	该工具将对启动阶段生成的输出进行解析，运用 SmartIO 算法计算最佳缓存大小。终端上将显示每个目标的最佳缓存大小、读取次数、延迟量、性能预测和缓存共享。

有关下载和安装工具的信息：

https://sort.veritas.com/dc_download/readme/smartassist

使用 SmartIO 功能：用例

本章节包括下列主题：

- 关于运行在 VxVM 卷上的应用程序的 SmartIO 读取缓存
- 关于对 VxFS 文件系统上运行的应用程序执行 SmartIO 读取缓存
- 关于使用 FSS 导出的 SSD 设备进行 SmartIO 缓存
- 关于对 VxFS 文件系统上运行的应用程序执行 SmartIO 写回缓存
- 关于在 VxFS 文件系统上执行读取和写回缓存的多个 SmartIO 缓存区域
- 关于 VxFS 文件系统上的 Oracle 数据库的 SmartIO 缓存
- 关于 VxVM 卷上的数据库的 SmartIO 缓存
- 技术预览：Veritas InfoScale 存储环境中的分布式 SmartIO

关于运行在 VxVM 卷上的应用程序的 SmartIO 读取缓存

对于 Veritas Volume Manager (VxVM) 卷，SmartIO 支持块级别读取缓存。此类 SmartIO 缓存主要支持直接运行在原始卷的应用程序，如直接运行在原始卷的数据库实例。卷级缓存还可以用在无法使用 VxFS 缓存的情况下。SmartIO 只支持卷级别的读取缓存。

SmartIO 缓存一般驻留在一个或多个 SSD 设备上或其他快速设备上。因为是从基于 SSD 的缓存（而不是标准存储系统）服务应用程序读取 I/O，所以 SmartIO 大幅提高了读取 I/O 的性能。

设置缓存时，SmartIO 不要求复杂的配置。设置缓存区很简单，它就被缓存的数据和关于缓存的元数据的存储空间。对于卷级别读取缓存，缓存区有 VxVM 类型。每个系统都有自己单独的 VxVM 缓存区。默认情况下，SmartIO 缓存区对这个系统上的所有 VxVM 卷启用自动缓存。需要的话，您可以将缓存区配置为 `noauto`。对

于 `noauto` 缓存区，您必须为 VxVM 卷显式启用 SmartIO 读取缓存。缓存区的配置是不变的。

请参见第 13 页的“[针对 VxVM 卷的自动缓存](#)”。

对于已启用缓存的每个 VxVM 卷，SmartIO 确定要缓存哪些数据或从缓存逐出哪些数据。SmartIO 借助于自身对工作负载的了解优化对缓存的使用。

SmartIO 功能只支持系统上的一个 VxVM 缓存区域。对于每个系统，被缓存的所有 VxVM 卷共享一个 VxVM 类型的缓存区。尽管同一个系统可能有 VxFS 缓存区和 VxVM 缓存区，但不支持多个 VxVM 缓存区。

缓存区对群集中的每个节点是专用的。缓存内容不会在该群集的节点间共享。

SmartIO 缓存保留卷级别的缓存一致性。如果缓存启用时缓存设备不可访问，此应用程序通常会继续运行。不过，应用程序的性能可能会有所减弱。

在 Cluster Volume Manager (CVM) 环境中，当写入操作写入到共享卷时，SmartIO 使用缓存一致性协议让多个节点上的缓存区域保持一致。在数据卷上写入会使其他节点的缓存区域上的内容变得无效。缓存一致性协议使用 Group Lock Manager (GLM) 模块进行通信。最初填充缓存时，缓存一致性协议会在写入 I/O 路径中创建很小的性能开销。

默认情况下，读取缓存中的数据不是持久性的。如果系统按计划发生重新启动，您能选择创建暖缓存。

请参见第 77 页的“[对持续或暖 VxVM 缓存的支持](#)”。

为 VxVM 卷使用 SmartIO 读取缓存所需的配置

对于具有以下配置的 VxVM 卷，可以设置 SmartIO 读取缓存：

- Storage Foundation RAC (SFRAC) 群集或 Storage Foundation Cluster File System High Availability (SFCFSA) 群集。缓存区不能位于共享卷上。必须将 VxVM 缓存区域配置为每个节点的本地缓存区域，因为不支持缓存区域的共享访问。
- Storage Foundation High Availability (SFHA) 群集。必须将 VxVM 缓存区域配置为每个节点的本地缓存区域，因为不支持缓存区域的共享访问。
请参见第 7 页的“[关于 SFHA 环境中的 SmartIO](#)”。
- 独立的 Storage Foundation 系统。

要缓存的卷的磁盘组版本必须为 190 或更高。

用于缓存区的设备具有以下特性：

- 利用 Veritas Volume Manager (VxVM) 支持的固态驱动器 (SSD) 等速度更快的设备提高读取 IO 的性能。但是，可以将 VxVM 支持的任何设备用于缓存区。
- 设备必须进行初始化以与 VxVM 配合使用，并且其格式必须为 `cdsdisk`。

针对 VxVM 卷的自动缓存

缓存区域的关联类型表示是否为系统启用自动缓存。VxVM 缓存区域的关联类型属性是持久性的。关联类型可以是下列类型之一：

- **auto** 属性（默认）
为自动缓存启用缓存区域。除非显式禁用卷的缓存，否则将缓存系统上的所有 VxVM 数据卷。您不需要在卷上显式启用缓存。
SmartIO 不支持缓存 RAID-5 卷和 DCO 卷。并且，SmartIO 不会为下列卷启用自动缓存：用于为包括 Storage Replication Log (SRL)、Data Change Map (DCM) 的对象记录日志和进行缓存的卷以及用于优化空间快照缓存对象的卷。默认情况下，VxVM 缓存区域具有 auto 属性。
- **noauto** 属性
不为自动缓存启用缓存区域。将不会自动缓存任何卷。您必须为每个需要缓存的卷显式启用缓存。您不需要显式禁用缓存的卷，除非要排除之前已启用的卷。您可以在创建卷时启用缓存。您也可以选择在某个现有的 VxVM 卷上启用或禁用读缓存，而无需请求 I/O。

为 VxVM 卷设置 SmartIO 读缓存

在读取模式下，SmartIO 功能可以缓存 VxVM I/O。要为 VxVM 卷读缓存设置 SmartIO，只需创建缓存区域。

为 VxVM 卷设置 SmartIO 读缓存

- 1 使用下列命令之一，在 SSD 设备上创建 VxVM 类型缓存区域：
 - 使用设备的磁盘访问名称 (*daname*) 指定一个或多个设备。设备应进行初始化，以供 VxVM 使用，并且其格式为 `cdsdisk`。

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

其中：

daname 指定要在其上创建缓存区域的设备的磁盘访问名称。

cachearea_name 指定缓存区域的自定义名称。如果未指定，SmartIO 功能会自动为缓存区域生成一个名称。

size 指定缓存区域的大小。默认情况下，缓存区会使用设备上的所有可用空间。

`--noauto` | `--auto` 指定缓存区的类型。默认值为 `--auto`。

`--nostripe` | `ncols=n` 指定缓存区域的布局选项。默认情况下，如果指定了两个或多个磁盘，则在条带卷上创建缓存区域。使用 `ncols=n` 选项指定条带卷的列数。使用 `--nostripe` 选项在指定磁盘的连续卷上创建缓存区域。

`cacheline_size` 指定 SmartIO 用于缓存的单位。当应用程序 I/O 访问数据时，SmartIO 会根据缓存块大小将数据移到缓存。您通常不需要更改 `cacheline_size`。

例如：

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

- 或者，您也可以指定要用于缓存区的磁盘组和卷的名称。在这种情况下，您不能指定大小，因此此时缓存区使用整个卷。

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

其中：

`dg/vol` 指定要用于缓存区域的磁盘组名称和卷名称。SmartIO 将此名称用于缓存区。

`--noauto|--auto` 指定缓存区的关联类型。默认值为 `--auto`。

例如：

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

- 2 为所需 VxVM 卷启用缓存。如果缓存区域是自动的，则无需此步骤。默认情况下，如果 SSD 设备联机，则为任何 VxVM 卷启用缓存。

如果缓存区域为 `noauto`，你必须为要缓存的卷启用缓存。

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

其中：

`dg/vol` 指定要缓存的卷的磁盘组名称和卷名称。

例如：

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

验证 VxVM 缓存区和监视缓存

配置了 SmartIO 功能之后，可以验证缓存区是否存在以及是否正在进行缓存。

验证并监视缓存区

- 1 使用以下命令可以显示有关系统上的缓存区的信息。

```
# 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
```

请参见第 61 页的“显示有关缓存区域的信息”。

2 使用以下命令可以显示有关某个特定缓存区的信息。

```
# 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
```

请参见第 71 页的[“查看 SmartIO 缓存统计数据”](#)。

3 要查看有关缓存使用情况的统计数据，请使用以下命令：

```
# sfcache stat sfcachearea_1
```

NAME	%CACHE	HIT RATIO		ART (Hit)ms		ART (Miss)ms		BYTES		
		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	

4 使用以下命令显示有关群集中其他节点使用的导出 SSD 的信息。在要查看现有缓存区域的节点上运行命令。

注意：其他节点的专用缓存区域并未列出。

```
# sfcache list --all
```

```
Hostname : sys1
NAME          TYPE  SIZE      ASSOC-TYPE  STATE  FSTYPE  DEVICE
sfcachearea_1 VxVM  10.00g    AUTO        ONLINE -        ibm_f90-0_0
sfcachearea_3 VxFS  10.00g    AUTO        ONLINE reserve ibm_f90-0_0

Hostname : sys2
NAME          TYPE  SIZE      ASSOC-TYPE  STATE  FSTYPE  DEVICE
sfcachearea_4 VxFS  20.00g    AUTO        ONLINE reserve ibm_f90-0_0
sfcachearea_5 VxVM  25.00g    AUTO        ONLINE -        ibm_f90-0_0

Hostname : sys3
NAME          TYPE  SIZE      ASSOC-TYPE  STATE  FSTYPE  DEVICE
sfcachearea_2 VxFS  10.00g    AUTO        ONLINE reserve ibm_f90-0_0
```

关于对 VxFS 文件系统上运行的应用程序执行 SmartIO 读取缓存

对于运行在 VxFS 文件系统上的应用程序，Storage Foundation High Availability Solutions 支持固态驱动器 (SSD) 上的读取缓存。在此情形中，将尽可能从缓存中满足应用程序读取操作。因为应用程序访问文件系统，文件系统从磁盘将数据加载到缓存。应用程序写入按常规方式转到磁盘。对于每一次写入，文件系统会同步缓存，保证应用程序不会看到失效的数据。如果缓存设备出了故障，在读取模式中

缓存的文件可能无法完整地呈现在这张磁盘上。所以，缓存故障不影响文件的应用程序 I/O，应用程序 I/O 会继续，不会中断。

为 VxFS 文件系统使用 SmartIO 读取缓存所需的配置

对于具有以下配置的 VxFS 文件系统，可以设置 SmartIO 读取缓存：

- 每个节点都具有专用 SSD 的 Storage Foundation RAC (SF Oracle RAC) 群集、Storage Foundation Cluster File System High Availability (SFCFSHA) 群集或 Storage Foundation High Availability (SFHA) 群集。
请参见第 7 页的“关于 SFHA 环境中的 SmartIO”。
- 独立的 Storage Foundation 系统。

文件系统必须具有以下特性：

- 单一缓存区域：文件系统布局版本 10。
多个缓存区域：文件系统布局版本 11。
- 必须在 VxVM 卷上装入。

VxFS 文件系统自动缓存

SmartIO 功能支持系统上的多个 VxFS 缓存区域。缓存区域的关联类型表示是否为系统启用自动缓存。VxFS 缓存区域的关联类型属性是持久性的。关联类型可以是下列类型之一：

- auto 属性（默认）
为自动缓存启用缓存区域。除非显式禁用对相应文件系统进行缓存，否则将对系统中的所有文件系统进行缓存。无需对文件系统显式启用缓存。默认情况下，VxFS 缓存区域具有 auto 属性。
- noauto 属性
不为自动缓存启用缓存区域。不自动缓存文件系统。必须针对想要缓存的每个文件系统显式启用缓存。无需对文件系统显式禁用缓存，除非是为了排除之前已启用的文件系统。

缓存区对群集中的每个节点是专用的。对于群集文件系统，群集中的每个节点都有其自己的缓存区域。缓存将基于每个节点进行，并且缓存内容将不会在群集中的各节点之间共享。已启用缓存的文件系统与每个节点上的本地缓存区域关联。

为 VxFS 文件系统设置 SmartIO 读缓存

在读取模式下，SmartIO 功能可以缓存 VxFS 文件系统读取 I/O。要为 VxFS 文件系统读缓存设置 SmartIO，只需创建缓存区域。

为 VxFS 文件系统设置 SmartIO 读缓存

1 使用以下命令之一在 SSD 设备上创建 VxFS 缓存区域。

- 使用设备的磁盘访问名称 (*daname*) 指定一个或多个设备。设备应进行初始化，以供 VxVM 使用，并且其格式为 `cdsdisk`。

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

其中：

daname 指定要在其上创建缓存区域的设备的磁盘访问名称。

cachearea_name 指定缓存区域的自定义名称。如果未指定，SmartIO 功能会自动为缓存区域生成一个名称。

size 指定缓存区域的大小。默认情况下，缓存区会使用设备上的所有可用空间。

`--nostripe|ncols=n` 指定缓存区域的布局选项。默认情况下，如果指定了两个或多个磁盘，则在条带卷上创建缓存区域。使用 `ncols=n` 选项指定条带卷的列数。使用 `--nostripe` 选项在指定磁盘的连续卷上创建缓存区域。

`--noauto|--auto` 指定缓存区的类型。默认值为 `--auto`。

例如：

```
# sfcache create ssd0_0
```

- 或者，您也可以指定要用于缓存区的磁盘组和卷的名称。在这种情况下，您不能指定大小，因此此时缓存区使用整个卷。

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

其中：

dg/vol 指定要用于缓存区域的磁盘组名称和卷名称。SmartIO 将此名称用于缓存区。

`--noauto|--auto` 指定缓存区的类型。默认值为 `--auto`。

例如：

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

2 如果没有装入文件系统，将装入 VxFS 文件系统。

- 如果缓存区域是自动的，则当您装入 VxFS 文件系统时会启用读缓存。例如，对于本地装入：

```
# mount -t vxfs /dev/vx/dsk/testdg/vol1 /mnt1
```

例如，对于 CFS 装入：

```
# mount -t vxfs -o cluster /dev/vx/dsk/testdg/vol1 /mnt1
```

- 如果缓存区域是非自动的，则必须为需要缓存的每个 VxFS 文件系统启用缓存。要启用缓存，请使用 `-o smartiomode` 选项装入该文件系统。例如，对于本地装入：

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

例如，对于 CFS 装入：

```
# mount -t vxfs -o cluster,smartiomode=read /dev/vx/dsk/testdg/vol1 /mnt1
```

装入文件系统后，您也可以启用缓存。

```
# sfcache enable mount_point
```

其中：

mount_point 是文件系统的装入点。

例如：

```
# sfcache enable /mnt1
```

- 3 如果需要，可以进一步自定义缓存行为。

请参见第 22 页的“[自定义缓存行为](#)”。

验证 VxFS 缓存区域并监视缓存

配置了 SmartIO 功能之后，可以验证缓存区是否存在以及是否正在进行缓存。

对于 VxFS 缓存区域，`sfcache list` 命令显示文件或目录的缓存模式。如果没有显示设置模式，文件或目录会继承装入点的缓存模式。如果显示设置了文件或目录的模式，会在重新装入后保留该值。显示的缓存模式可能与为装入点启用的模式有所不同。除非文件系统在 `writeback` 模式下装入，否则不会启用 `writeback` 模式。如果文件或目录设置为 `writeback` 模式，但文件系统在其他模式下装入，则文件或目录会继承装入点的缓存模式。

验证并监视缓存区

- 1 使用以下命令可以显示有关系统上的缓存区的信息。

```
# sfcache list
```

```
NAME          TYPE SIZE  ASSOC-TYPE  STATE  FSTYPE  DEVICE
sfcachearea_2 VxFS 7.00g AUTO          ONLINE default sdg
```

- 2 使用以下命令可以显示有关某个特定缓存区的信息。

```
# sfcache list sfcachearea_2
```

```
Cachearea: sfcachearea_2
```

```
Assoc Type: AUTO
```

```
Type: VxFS
```

```
Size: 9.96g
```

```
State: ONLINE
```

```
Layout: CONCAT
```

```
FStype: default
```

```
Number of Columns: 0
```

```
/dev/vx/dsk/sfcache_defaultdg/sfcachearea_2:
```

FSUUID	SIZE	MODE	MOUNTPOINT	CACHENAME
a6178a5604a50200577a0000a...	759.0 MB	read	/ora_inst1	sfcachearea_2
0d929856a7d60e003d380000f...	512.0 MB	writeback	/pdb2	sfcachearea_2
90929856b688000081410000b...	50.1 MB	read	/fast_recovery	sfcachearea_2
02188a569d7e0700567d00007...	4 KB	nocache	-	sfcachearea_2
31188a5679a90900987d00007...	4 KB	nocache	-	sfcachearea_2
a55b8a56db160a00fa610000f...	4 KB	nocache	-	sfcachearea_2

- 3 显示有关特定文件系统的信息：

```
# 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 要查看有关缓存使用情况的统计数据，请使用以下命令：

```
# sfcache stat /mnt1
```

```
TYPE: VxFS :
    Cache Name: sfcachearea_1
    Cache Size:      5 GB
Cache Utilization:      4 KB ( 0.00 %)

Read Cache                               Writeback

Hit Ratio Data Read Data Written Hit Ratio Data Written rdcachename wbcachename

/mnt1:
    0.00 %  0 KB      0 KB          0.00 %   0 KB          sfcachearea_1 sfcachearea_2
```

输出显示缓存数据的统计信息。

请参见第 71 页的“[查看 SmartIO 缓存统计数据](#)”。

- 5 使用以下命令显示有关群集中其他节点使用的导出 SSD 的信息。在要查看现有缓存区域的节点上运行命令。

注意：其他节点的专用缓存区域并未列出。

```
# sfcache list --all
```

```
Hostname : sys1
NAME      TYPE  SIZE  ASSOC-TYPE STATE  FSTYPE  DEVICE
sfcachearea_1 VxVM  10.00g AUTO   ONLINE -        ibm_f90-0_0
sfcachearea_3 VxFS  10.00g AUTO   ONLINE reserve ibm_f90-0_0

Hostname : sys2
NAME      TYPE  SIZE  ASSOC-TYPE STATE  FSTYPE  DEVICE
sfcachearea_4 VxFS  20.00g AUTO   ONLINE reserve  ibm_f90-0_0
sfcachearea_5 VxVM  25.00g AUTO   ONLINE -        ibm_f90-0_0

Hostname : sys3
NAME      TYPE  SIZE  ASSOC-TYPE STATE  FSTYPE  DEVICE
sfcachearea_2 VxFS  10.00g AUTO   ONLINE reserve  ibm_f90-0_0
```

自定义缓存行为

默认情况下，SmartIO 根据工作负载缓存文件数据。SmartIO 根据 I/O 访问情况将一部分文件加载到缓存中。当缓存区域填满时，数据可能会被逐出，以便为缓存新

数据腾出空间。SmartIO 会根据条件（例如访问频率）逐出数据。当数据位于缓存中时，将从缓存提供该文件数据的后续 I/O。如果数据已被逐出，将由主存储来满足所有后续 I/O 请求。然后 SmartIO 可能会重新缓存该数据。

为了最大程度地利用缓存，可以通过自定义缓存行为来控制何时从缓存加载或逐出文件。可以通过以下操作自定义缓存行为：

- `load` 操作可在 I/O 访问文件之前先将文件预加载到缓存中。文件已经在缓存中，I/O 便可以更快速地返回。默认情况下，文件加载在后台进行。使用 `-o sync` 操作可同步加载文件，即指定该命令只有在所有文件均加载完之后才返回。以此方式加载的文件受常规的逐出条件限制。
- `pin` 操作可防止文件被从缓存中逐出。可以固定常用文件，以使 SmartIO 不会逐出这些文件以后再重新进行缓存。已固定的文件将无限期地保留在缓存中，直到该文件被删除或已明确被取消固定。如果使用 `-o load` 选项固定文件，则该操作还将同步缓存文件内容。如果没有指定 `-o load` 选项，则将根据 I/O 访问情况缓存文件内容。
- `unpin` 操作可删除文件的固定状态。`unpin` 操作不会立即逐出该文件。当缓存中需要空间时，SmartIO 会考虑按照对任何其他文件的相同方式逐出该文件。

对于上述每个操作，可以分别指定各个文件，也可以指定目录名对目录中的所有文件进行相应处理。使用 `-r` 选项进行递归选择。

加载文件或目录

- ◆ 要将某文件或目录加载到缓存中，请在以下命令中指定一个或多个文件名或目录名。

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

使用 `-r` 选项进行递归选择。

使用 `-o sync` 选项指定命令在所有文件均加载完之后才返回。

固定文件或目录

- ◆ 要将某文件或目录固定到缓存中，请在以下命令中指定一个或多个文件名或目录名。

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

使用 `-r` 选项进行递归选择。

使用 `-o load` 选项将文件同步加载到缓存中。

取消固定文件或目录

- ◆ 要从缓存中取消固定某文件或目录，请在以下命令中指定一个或多个文件名或目录名。

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

使用 `-r` 选项进行递归选择。

关于使用 FSS 导出的 SSD 设备进行 SmartIO 缓存

SmartIO 支持使用由 FSS 导出的固态驱动器 (SSD) 为 Veritas Volume Manager (VxVM) 和 Veritas File System (VxFS) 上运行的应用程序提供缓存服务。在这种情况下，Flexible Storage Sharing (FSS) 从具有本地 SSD 的节点中导出 SSD。然后，FSS 将在群集中创建导出的 SSD 池。通过该共享池为群集中的任何或所有节点创建缓存区域。每个缓存区域只能由为其创建缓存区域的特定节点访问。该缓存区域可以是 VxVM 缓存区域，也可以是 VxFS 缓存区域。

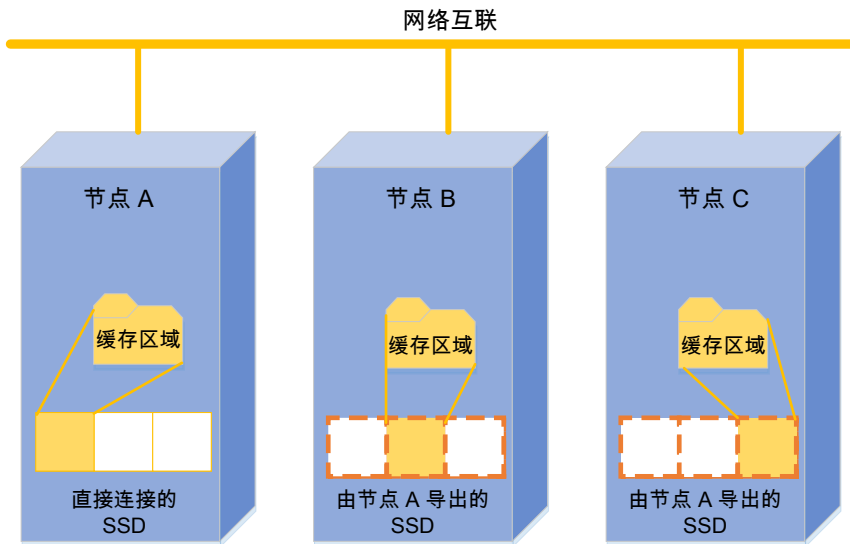
SmartIO 支持对 FSS 导出的远程 SSD 设备上的本地装入执行写回缓存。但是，不支持对 CFS 环境的远程 SSD 设备执行写回缓存。

如果计划仅使用部分导出 SSD 设备执行缓存，请确保在 200 或更高版本的磁盘组上创建缓存所使用的卷。

远程 SSD 上缓存区域的卷布局遵循简单条带布局，而不是主机上默认的 FSS 镜像分配策略。如果需要调整远程 SSD 上缓存区域的大小以满足增长需求，请确保仅指定一个导出的设备。如果指定了非导出设备，则操作会失败。重新启动后，可以启用该缓存区域以支持暖缓存或持续缓存。

[图 2-1](#) 说明了该缓存配置。

图 2-1 使用 FSS 导出的 SSD 进行 SmartIO 缓存



节点退出或加入群集时缓存区域的状态

表 2-1 介绍了节点退出或加入群集时缓存的状态。

表 2-1 节点退出或加入群集时缓存区域的状态

方案	缓存状态
使用远程缓存的节点退出群集	节点退出群集后，由于缓存区域无法访问，因此将禁用缓存。
不具有本地 SSD 的节点加入群集	节点加入群集时，VxVM 会使缓存区域联机。根据缓存区域的关联类型，为卷启用缓存。如果关联类型设置为 auto，将启用自动缓存。如果关联类型设置为 noauto，将禁用自动缓存。
提供存储的节点退出群集	使用该节点中的磁盘创建的所有缓存区域都将分离，并且无法访问。群集中的剩余节点将无法访问这些磁盘。节点重新加入群集时，将重新启动缓存。
具有本地 SSD 的节点加入群集	节点加入群集时，使用该节点中的存储创建的所有缓存区域都将联机。

使用 FSS 导出的 SSD 设置缓存区域

登录到要从中导出 SSD 设备的节点。然后，使用导出的 SSD 创建缓存区域。

使用 FSS 导出的 SSD 设置缓存区域

- 1 登录到要从中导出 SSD 的节点。
- 2 初始化磁盘以与 VxVM 配合使用：

```
# vxdisk init disk_name
```

- 3 从主机导出 SSD 设备：

```
# vxdisk export disk_name disk_name
```

- 4 登录到要使用在步骤 3 中导出的设备为其创建缓存区域的节点。
- 5 使用导出的 SSD 创建 VxVM 或 VxFS 缓存区域：

```
# sfcache create [-t cache_type] [cachearea_name] \  
{ssd_device} [size]
```

- 6 查看缓存区域：

```
# sfcache list --all
```

关于对 VxFS 文件系统上运行的应用程序执行 SmartIO 写回缓存

对于运行在 Veritas File System (VxFS) 文件系统上的应用程序，Storage Foundation and High Availability Solutions 支持固态硬盘 (SSD) 上的写回缓存。在此情形中，将尽可能从缓存中满足应用程序读写操作。

注意：当前 SF Oracle RAC 环境不支持 SmartIO 写回缓存。

SmartIO 提供 writeback 模式的写入缓存。在 writeback 模式下，在此数据写入到 SmartIO 缓存（通常是在 SSD 上）后，应用程序写入操作会返回成功信息。以后，SmartIO 会刷新缓存，将脏数据写到磁盘。写回缓存期望改进同步用户数据写入的延迟。将脏数据刷新到磁盘时，无法保证写入顺序保真。

写回缓存是读取缓存的超集。启用写回缓存时，将隐式启用读取缓存。将尽可能从缓存中满足读取操作，并且文件系统会透明地将文件数据加载到缓存。可以同时为同一个文件启用读取和写回缓存这两项操作。

`writeback` 缓存模式提供了性能良好的写入操作，但也意味着磁盘副本可能不总是最新的。如果缓存设备出了故障，在 `writeback` 模式中缓存的文件可能无法完整地呈现在这张磁盘上。当设备重新联机时，SmartIO 有从缓存设备刷新数据的机制。Storage Foundation Cluster File System High Availability (SFCFSHA) 利用缓存反射，为避免数据丢失提供了额外的保护。

在 SFCFSHA 的情况下，当启用 `writeback` 模式的缓存时，SmartIO 会将文件系统级别的写回数据镜像到其他节点的 SSD 缓存。这一被称为缓存反射的行为可防止在节点出现故障的情况下发生写回数据丢失。如果一个节点发生故障，作为重新配置的一部分，另一个节点会刷新这个故障节点的镜像脏数据。即使节点出现故障，存在暂停的脏数据，缓存反射也能保证写回数据没有丢失。

在本地装入的情况下，如果没有缓存镜像，在此节点发生故障时，这个文件的磁盘副本可能是不完整或失效的。

装入点启用写回缓存后，该文件系统中合格的同步写入将被缓存。SmartIO 使用类似以下的标准来确定写入是否符合写回缓存的资格：

- 写入请求必须与 `PAGESIZE` 一致（4k 的倍数）。
- 写入请求不能大于 2MB。
- 发生写入操作的文件没有 `mmap`。
- `writeback` 模式的缓存没有被管理员显式禁用。

通过添加参数信息帮助 SmartIO 功能进行决策，您能对缓存哪些数据进行自定义。

为 VxFS 文件系统使用 SmartIO 写回缓存所需的配置

对于 Storage Foundation 或 Storage Foundation High Availability，必须拥有 Enterprise 许可证才能使用 SmartIO 的写回缓存功能。

对于具有以下配置的 VxFS 文件系统，可以设置 SmartIO 写回缓存：

- 恰好具有 2 个节点的 Storage Foundation Cluster File System High Availability (SFCFSHA) 群集。如果群集具有 2 个以上节点，写回缓存不会启用。配置 `writeback` 模式缓存时如果添加了另一个节点，则写回缓存会被禁用。此时缓存会一直处于读取模式。
配置了 `writeback` 模式缓存时，如果在两个节点的其中一个节点上卸载了群集文件系统，则写回缓存会被禁用。此时缓存会一直处于读取模式。如果在第二个节点上重新装入了群集文件系统，写回缓存会自动启用。
- 本地装入配置。

对于 CFS，写回缓存会使用 LLT 传输来创建写回数据的镜像。写回前，缓存的应用程序写入内容还会写入到远程缓存。

Veritas 建议在 10GigE 或 RDMA 等高带宽网络上配置 LLT，以免影响吞吐量。

有关配置 LLT 的信息，请参见《Storage Foundation Cluster File System High Availability 配置和升级指南》。

文件系统必须具有以下特性：

- 文件系统对单一缓存区域使用磁盘布局版本 10 或更高版本，对多个缓存区域使用版本 11。
- 必须在 VxVM 卷上装入。

为 VxFS 文件系统设置 SmartIO 写回缓存

在 `writeback` 模式下，SmartIO 功能会缓存 VxFS 文件系统的读写 I/O。要为 VxFS 文件系统设置 SmartIO 写回缓存，请在 `writeback` 模式下创建缓存区域并装入文件系统。

为 VxFS 文件系统设置 SmartIO 写回缓存

1 使用以下命令之一在 SSD 设备上创建 VxFS 缓存区域。

- 使用设备的磁盘访问名称 (*daname*) 指定一个或多个设备。设备应进行初始化，以供 VxVM 使用，并且其格式为 `cdsdisk`。

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

其中：

daname 指定要在其上创建缓存区域的设备的磁盘访问名称。

cachearea_name 指定缓存区域的自定义名称。如果未指定，SmartIO 功能会自动为缓存区域生成一个名称。

size 指定缓存区域的大小。默认情况下，缓存区会使用设备上的所有可用空间。

`--noauto` | `--auto` 指定缓存区的类型。默认值为 `--auto`。

例如：

```
# sfcache create ssd0_0
```

`--nostripe|ncols=n` 指定缓存区域的布局选项。默认情况下，如果指定了两个或多个磁盘，则在条带卷上创建缓存区域。使用 `ncols=n` 选项指定条带卷的列数。使用 `--nostripe` 选项在指定磁盘的连续卷上创建缓存区域。

- 或者，您也可以指定要用于缓存区的磁盘组和卷的名称。在这种情况下，您不能指定大小，因此此时缓存区使用整个卷。

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

其中：

`--noauto` | `--auto` 指定缓存区的类型。默认值为 `--auto`。

`dg/vol` 指定要用于缓存区域的磁盘组名称和卷名称。SmartIO 将此名称用于缓存区。
 例如：

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

- 2 装入 VxFS 文件系统并将 `smartiomode` 选项设置为 `writeback`。如果文件系统已装入，则必须重新装入文件系统以将 `smartiomode` 选项设置为 `writeback`。

本地装入示例：

```
# mount -t vxfs -o smartiomode=writeback /dev/vx/dsk/testdg/vol1 \
/mnt1
```

重新装入示例：

```
# mount -t vxfs -o remount,smartiomode=writeback \
/dev/vx/dsk/testdg/vol1 /mnt1
```

对于群集文件系统，必须同时在两个节点上装入文件系统，并且必须设置 `smartiomode` 和 `cluster` 选项。

当您使用这些选项装入 CFS 文件系统时，SmartIO 会自动在其他节点的 SSD 缓存中反映缓存。

CFS 装入示例：

```
# mount -t vxfs -o cluster,smartiomode=writeback \
/dev/vx/dsk/testdg/vol1 /mnt1
```

要启用写回缓存，无论缓存区域为 `auto` 还是 `noauto`，都必须将 `smartiomode` 选项设置为 `writeback`。如果缓存区域为 `auto` 并且未设置 `smartiomode`，则会以默认的读取模式为文件系统启用 SmartIO 缓存。

- 3 如果需要，可以进一步自定义缓存行为。

请参见第 22 页的“自定义缓存行为”。

请参见第 70 页的“调整写回缓存”。

验证 VxFS 缓存区域并监视缓存（写回模式）

配置了 SmartIO 功能之后，可以验证缓存区是否存在以及是否正在进行缓存。

对于 VxFS 缓存区域，`sfcache list` 命令显示文件或目录的缓存模式。如果没有显示设置模式，文件或目录会继承装入点的缓存模式。如果显示设置了文件或目录的模式，会在重新装入后保留该值。显示的缓存模式可能与为装入点启用的模式有所不同。除非文件系统在 `writeback` 模式下装入，否则不会启用 `writeback` 模式。

如果文件或目录设置为 `writeback` 模式，但文件系统在其他模式下装入，则文件或目录会继承装入点的缓存模式。

验证并监视缓存区

1 显示有关系统上缓存区域的信息

```
# sfcache list
```

例如，单节点 VxFS 缓存区域显示的输出如下：

```
NAME                TYPE SIZE  ASSOC-TYPE  STATE  FSTYPE  DEVICE
sfcachearea_2      VxFS 7.00g AUTO           ONLINE default  sdg
```

对于已启用 `writeback` 的群集文件系统，如果您在已启用 `writeback` 的情况下刚刚装入后（并且尚未出现任何读取缓存）发出 `sfcache list` 命令，输出将缓存中使用的空间显示为 **1.0 GB**。缓存反射已使用每个日志大小为 **512 MB** 的本地日志和远程日志配置。

2 显示有关特定缓存区域的信息：

```
# sfcache list sfcachearea_2
```

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

```
/dev/vx/dsk/sfcache_defaultdg/sfcachearea_2:
FSUUID                SIZE      MODE      MOUNTPOINT  CACHENAME
a6178a5604a50200577a0000a... 759.0 MB  read      /ora_inst1  sfcachearea_2
0d929856a7d60e003d380000f... 512.0 MB  writeback /pdb2       sfcachearea_2
90929856b688000081410000b... 50.1 MB   read      /fast_recovery sfcachearea_2
02188a569d7e0700567d00007... 4 KB     nocache   -           sfcachearea_2
31188a5679a90900987d00007... 4 KB     nocache   -           sfcachearea_2
a55b8a56db160a00fa610000f... 4 KB     nocache   -           sfcachearea_2
```

输出显示有关缓存配置和属性的信息。

请参见第 61 页的“[显示有关缓存区域的信息](#)”。

3 显示有关特定文件系统的信息：

```
# 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 要查看有关缓存使用情况的统计数据，请使用以下命令：

```
# sfcache stat sfcachearea_1
```

```
TYPE: VxFS
NAME: sfcachearea_1
  Cache Name: sfcachearea_1
    Cache Size:      5 GB
    Cache Utilization: 1.426 GB (28.51 %)
File Systems Using Cache:      2
Writeback Cache Use Limit: Unlimited
  Writeback Flush Timelag:    10 s

Read Cache                                Writeback
Hit Ratio  Data Read  Data Written  Hit Ratio  Data Written rdcachename  wbcachename

Total:
  7.98 %  157.7 MB  1.545 GB    0.00 %    0 KB

/pdb1:
  7.98 %  157.7 MB  1.545 GB    0.00 %    0 KB          sfcachearea_1  -

/pdb2:
  0.00 %  0 KB      0 KB        0.00 %    0 KB          sfcachearea_1  sfcachearea_2
```

输出显示缓存数据的统计信息。

请参见第 71 页的“[查看 SmartIO 缓存统计数据](#)”。

5 要查看有关特定文件系统的缓存使用情况统计信息，请使用以下命令：

```
# sfcache stat /mnt1

TYPE: VxFS :
    Cache Name: sfcachearea_1
    Cache Size:      5 GB
Cache Utilization:    4 KB ( 0.00 %)

Read Cache                                Writeback

Hit Ratio Data Read Data Written Hit Ratio Data Written rdcachename wbcachename

/mnt1:
    0.00 %  0 KB      0 KB          0.00 %   0 KB          sfcachearea_1 sfcachearea_2
```

6 检查 syslog 以验证是否已启用 writeback 模式的缓存。

在 syslog 中应该会看到如下行：

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

如果特定文件系统的 writeback 模式缓存处于禁用状态，则在 syslog 中会看到如下行：

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

关于在 VxFS 文件系统中执行读取和写回缓存的多个 SmartIO 缓存区域

您可以运行 VxFS 文件系统在单节点系统和多节点系统上创建多个缓存区域。可以对读取操作和写回操作使用不同的缓存区域。每个应用程序均可对其读写操作使用一个缓存区域，对写回操作使用另一个缓存区域。您可以配置将同一缓存区域作为某些应用程序的读取缓存并作为另外一些应用程序的写回缓存，

VxFS 支持下列类型的缓存区域：

默认值

适用于除配置为使用保留的缓存区域的应用程序以外的所有应用程序。

每次只能有一个默认缓存区域联机。强制要求设置默认缓存区域。

Reserved (保留)

保留供某些应用程序使用。仅适用于配置为使用缓存区域的应用程序。一个或多个应用程序可使用同一缓存区域作为保留缓存区域。

可以设置任意数量的保留缓存区域。

可以在创建缓存区域时设置缓存类型。如果未指定缓存区域，则将缓存区域设置为“default (默认)”。可以将缓存区域从一种类型转换为另一种类型。如果未为应用程序定义缓存区域，则使用默认缓存区域。

图 2-2 说明多个 SmartIO 缓存区域上的读取和写回缓存。

图 2-2 多个 SmartIO 缓存区域上的读取和写回缓存

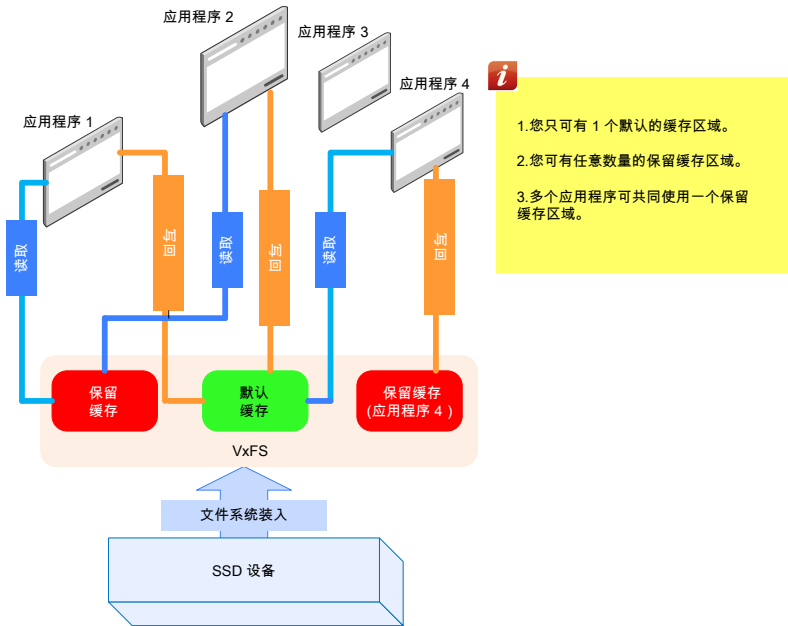


图 2-3 说明在单个节点系统上创建多个 SmartIO 缓存区域的步骤。

图 2-3 在单个节点系统上创建多个 SmartIO 缓存区域的步骤

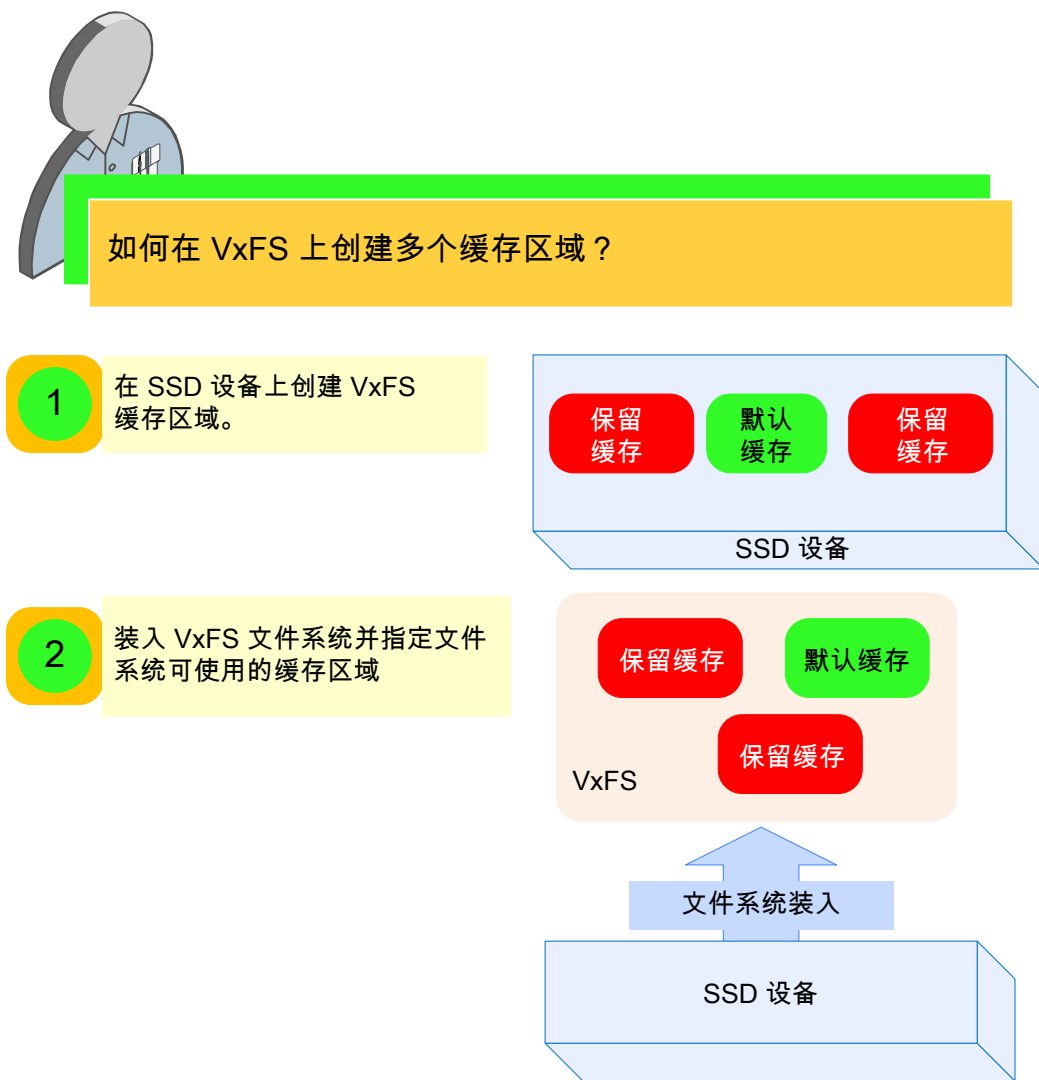


图 2-4 说明在多个节点群集上创建多个 SmartIO 缓存区域的步骤。

图 2-4 在多个节点群集上创建多个 SmartIO 缓存区域的步骤



如何在 CFS 上创建多个缓存区域？

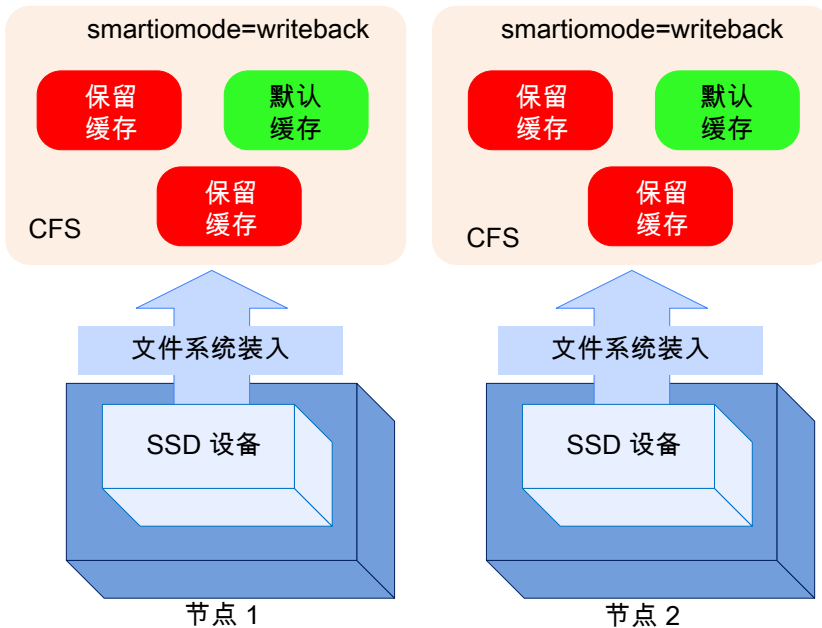
1

在 SSD 设备上创建 VxFS 缓存区域。



2

装入 VxFS 文件系统并指定文件系统可使用的缓存区域



i

smartiomode 必须在群集中的所有节点上保持相同并且拥有下列模式之一：

- writeback
读取和写回缓存操作
- read
仅读取缓存操作
- nocache
无缓存操作

关于 smartiocache 选项

mount 命令的 smartiocache 选项可以指定可供文件系统使用的缓存区域。可以为文件系统上的读取和写回缓存指定不同的缓存区域。此选项要求文件系统安装磁盘布局版本 11 及更高版本。

```
mount -t vxfs -o
smartiomode=[read|writeback],smartiocache=[cachearea_name]
\[ [rdcachearea_name]:[wbcachearea_name]] file_system \mount_point
```

其中：

cachearea_name 指定 SmartIO 缓存区域的名称。

rdcachearea_name 指定读取缓存的缓存区域的名称。

wbcachearea_name 指定写回缓存的缓存区域的名称

- smartiocache 的值取决于为 smartiomode 指定的值
- 可以在重新装入文件系统时更改缓存区域名称。指定新缓存区域时，现有缓存区域将与文件系统分离。
如果装入文件系统时指定的缓存区域未联机，则无法使用缓存区域启动文件系统。联机后，文件系统可以使用缓存区域。
- 如果仅指定一个缓存区域名称并将 smartiomode 设置为 writeback，将使用该缓存区域执行读取和写回缓存操作。
如果未指定缓存区域名称且在自动模式下默认缓存区域为联机状态，则使用默认缓存区域根据 smartiomode 设置执行缓存。
- 如果使某个装入点的读取缓存区域脱机，则还将禁用该装入点的写回缓存。启用读取缓存后又会再次启用。

将 VxFS 缓存区域从一种类型转换为另一个类型

您可以在缓存处于联机状态时，使 VxFS 缓存区域在保留状态与默认状态之间切换。

当发生以下情况时，缓存转换将失败：

- 缓存脱机。
- 如果默认缓存区域已存在，保留缓存区域将转换为默认缓存区域。VxFS 仅支持在一个系统上包含一个默认联机缓存区域。

例如，如果缓存区域 cache1 为保留类型，并且希望将该缓存区域作为默认缓存区域，请运行以下命令：

```
# sfcache set type=default cache1
```

如果缓存区域 `cache2` 为默认类型，并且希望为某个特定应用保留该缓存区域，请运行以下命令：

```
# sfcache set type=reserve cache2
```

在系统上设置多个缓存区域

首先，创建缓存区域。然后，装入文件系统并指定可供文件系统使用的缓存区域。

在系统上设置多个缓存区域

1 使用以下命令之一在 SSD 设备上创建 VxFS 缓存区域。

- 使用设备的磁盘访问名称 (*daname*) 指定一个或多个设备。设备应进行初始化，以供 VxVM 使用，并且其格式为 `cdsdisk`。

```
# sfcache create [-t VxFS] [size]  
daname[...] [--auto|--noauto] \  
[--default|--reserve] [cachearea_name]
```

其中：

daname 指定要在其上创建缓存区域的设备的磁盘访问名称。

cachearea_name 指定缓存区域的自定义名称。如果未指定，SmartIO 功能会自动为缓存区域生成一个名称。

size 指定缓存区域的大小。默认情况下，缓存区会使用设备上的所有可用空间。

`--noauto|--auto` 指定缓存区域模式。默认值为 `--auto`。

`--default|--reserve` 指定缓存区域的类型。默认值为 `--default`

例如：

```
# sfcache create ssd1 --reserve cache1
```

- 或者，您也可以指定要用于缓存区的磁盘组和卷的名称。在这种情况下，您不能指定大小；缓存区域使用整个卷。

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

其中：

`--noauto|--auto` 指定缓存区域模式。默认值为 `--auto`。

`--default|--reserve` 指定缓存区域的类型。默认值为 `--default`

dg/vol 指定要用于缓存区域的磁盘组名称和卷名称。SmartIO 将此名称用于缓存区。

例如：

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

- 2 装入 VxFS 文件系统并指定可供文件系统使用的缓存区域。

在以下示例中，*cache1* 用于读取缓存，*cache2* 用于写回缓存：

```
# mount -t vxfs -o smartiomode=writeback,smartiocache=cache1:cache2 \
/dev/vx/dsk/testdg/vol1 \
/mnt1
```

在以下示例中，*cache1* 用于读取缓存：

```
# mount -t vxfs -o smartiomode=read,smartiocache=cache1 \
/dev/vx/dsk/testdg/vol1 /mnt1
```

在以下示例中，*cache1* 用于读取缓存及写回缓存：

```
# mount -t vxfs -o smartiomode=writeback,smartiocache=cache1 \
/dev/vx/dsk/testdg/vol1 /mnt1
```

对于群集文件系统，必须同时在两个节点上装入文件系统，并且必须设置 `cluster` 和 `smartiomode` 选项。`smartiomode` 在群集的所有节点上必须相同。

CFS 装入示例：

```
# mount -t vxfs -o cluster,smartiomode=writeback,\
smartiocache=cache1:cache2 /dev/vx/dsk/testdg/vol1 /mnt1
```

如果存在默认缓存，且未指定 `smartiocache` 选项，VxFS 将对指定的缓存模式自动使用默认缓存。

```
# mount -t vxfs -o cluster,smartiomode=writeback,\
/dev/vx/dsk/testdg/vol1 /mnt1
```

- 3 如果需要，可以进一步自定义缓存行为。

请参见第 22 页的“自定义缓存行为”。

请参见第 70 页的“调整写回缓存”。

验证 VxFS 缓存区域

配置缓存区域后，验证缓存区域是否存在以及是否正在进行缓存。

验证 VxFS 缓存区域

1 显示有关系统上缓存区域的信息

```
# sfcache list
```

例如，单节点 VxFS 缓存区域显示的输出如下：

NAME	TYPE	SIZE	ASSOC-TYPE	STATE	FSTYPE	DEVICE
sfcachearea_1	VxFS	5.00g	AUTO	ONLINE	reserve	ssd0_0
sfcachearea_2	VxFS	7.00g	AUTO	ONLINE	default	ssd0_1

对于已启用 writeback 的群集文件系统，如果您在已启用 writeback 的情况下刚刚装入后（并且尚未出现任何读取缓存）发出 sfcache list 命令，输出将缓存中使用的空间显示为 1.0 GB。缓存反射已使用每个日志大小为 512 MB 的本地日志和远程日志配置。

2 显示有关特定缓存区域的信息：

```
# sfcache list sfcachearea_2
```

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

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

FSUUID	SIZE	MODE	MOUNTPOINT	CACHENAME
5efe4a52eb76000041760000a0dec33fe70300005efe4a52	39.0MB	read	/mnt2	sfcachearea_2

输出显示有关缓存配置和属性的信息。

请参见第 61 页的“显示有关缓存区域的信息”。

3 显示有关特定文件系统的信息：

```
# 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 要查看有关缓存使用情况的统计数据，请使用以下命令：

```
# sfcache stat sfcachearea_1

TYPE: VxFS
NAME: sfcachearea_1
    Cache Name: sfcachearea_1
        Cache Size:      5 GB
        Cache Utilization: 1.426 GB (28.51 %)
File Systems Using Cache:      2
Writeback Cache Use Limit: Unlimited
    Writeback Flush Timelag:    10 s

Read Cache                                Writeback
Hit Ratio  Data Read  Data Written Hit Ratio  Data Written
rdcachename  wbcachename

Total:
    7.98 %  157.7 MB   1.545 GB    0.00 %    0 KB

/pdb1:
    7.98 %  157.7 MB   1.545 GB    0.00 %    0 KB
sfcachearea_1      -

/pdb2:
    0.00 %  0 KB         0 KB        0.00 %    0 KB
sfcachearea_1  sfcachearea_2
```

输出显示缓存数据的统计信息。

请参见第 71 页的“[查看 SmartIO 缓存统计数据](#)”。

5 要查看有关特定文件系统的缓存使用情况统计信息，请使用以下命令：

```
# sfcache stat /mnt1
```

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

Read Cache                Writeback
Hit Ratio  Data Read Data Written Hit Ratio Data Written rdcachename wbcachename

/mnt1:
  0.00 %  0 KB          78.0 MB      100.00 %   39.0 MB   sfcachearea_1 sfcachearea_2
```

6 检查 `syslog` 以验证是否已启用 `writeback` 模式的缓存。

在 `syslog` 中应该会看到如下行：

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

如果特定文件系统的 `writeback` 模式缓存处于禁用状态，则在 `syslog` 中会看到如下行：

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

关于 VxFS 文件系统上的 Oracle 数据库的 SmartIO 缓存

SmartIO 提供了应用程序模板，以优化运行在 VxFS 文件系统上的数据库的缓存。SmartIO 使用这些模板，对此数据库中特殊类型的信息应用策略。例如，索引文件可能有与数据文件不同的缓存策略。

SmartIO 为 VxFS 文件系统提供下列应用程序模板：

- 模板名：`oracle`
优化运行 VxFS 文件系统上的 Oracle 数据库的缓存。

对 Oracle 使用 SmartIO 插件的先决条件和配置

Storage Foundation and High Availability Solutions 的 SmartIO 功能包括一个用于 Oracle 数据库的插件。

适用于 Oracle 的 SmartIO 插件需要 Oracle 11 或更高版本。该限制不适用于没有插件的 SmartIO 缓存。

确保正确地配置了此系统，以便为 Oracle 使用 SmartIO 插件。

配置此系统，以便为 Oracle 使用 SmartIO 插件

- 1 执行 Oracle SmartIO 插件前，请创建文件系统缓存区域并且将缓存区域联机。
- 2 Oracle SmartIO 插件需要查询数据库目录表。确保 Oracle 数据库已联机，并且运行在您需要运行 Oracle 插件的 `sfcache` 命令的同一台主机上。
- 3 为使 `sfcache app oracle` 命令有效，`/etc/oratab` 文件必须存在并且必须包括以下行：

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

其中：

`oraclesid` 是服务器上 Oracle 实例的系统 ID (SID)。

`oracle_home` 是与该实例相关联的 `ORACLE_HOME` 目录。

`Y|N` 标志表示该实例是否应该在引导时自动启动。

为 VxFS 文件系统上运行的数据库设置默认的 SmartIO 缓存策略

SmartIO 提供了应用程序模板，以优化运行在 VxFS 文件系统上的数据库的缓存。SmartIO 使用模板对数据库中特定类型的文件应用策略。例如，针对索引文件和数据文件的缓存策略。

Oracle 模板为 Oracle 数据库设置了默认策略，如下所示：

- 对 ARCHLOG 文件关闭缓存（非缓存模式）
- 对 TEMPFILE 设置读取缓存（读取模式）
- 对于 OLTP 数据库，为读取频率最高的数据文件设置读取缓存。
对于 OLAP 数据库，为包含 INDEX 的所有数据文件设置读取缓存。

要为数据库设置默认的 SmartIO 缓存策略，请以 ORACLE 用户身份运行下列命令：

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

其中：

`$ORACLE_HOME` 和 `$ORACLE_SID` 是必需的，用于唯一标识数据库。

OLAP 或 OLTP 表示应用程序负载类型。OLAP，即“联机分析处理”应用程序处理用于多维分析查询的工作负载。OLTP，即“联机事务处理”应用程序处理面向事务的工作负载，例如数据录入和检索事务处理。

OLTP 数据库示例：

```
$ 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 数据库示例:

```

$ 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
```

设置数据库对象的 SmartIO 缓存策略

了解数据库活动和使用统计数据的数据管理员 (DBA) 可能希望根据此信息来调整 SmartIO 缓存策略。您可以设置指定数据库对象（包括已命名表空间、最新分区或特定数据文件）的 SmartIO 缓存策略。您也可以定位指定数据库对象，将其保存在 SmartIO 缓存区域中。

请参见第 45 页的“固定和取消固定数据库对象”。

设置指定数据库对象的缓存策略

- ◆ 请使用以下命令：

```
# 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}
```

固定和取消固定数据库对象

使用此过程固定或取消固定指定数据库对象，包括已命名表空间、表群集、分区或数据文件。

- `pin` 操作可防止数据从缓存中被逐出。可以固定常用数据库对象，以使 SmartIO 不会逐出这些数据以后再重新进行缓存。已固定的数据将无限期地保留在缓存中，直到数据被删除或已明确被取消固定。
- `unpin` 操作可使数据脱离固定状态。`unpin` 操作不会立即逐出该数据。当缓存中需要空间时，SmartIO 会考虑按照对任何其他数据采取的相同方式逐出该数据。

固定或取消固定指定数据库对象

- ◆ 要固定或取消固定指定数据库对象（包括已命名表空间、表群集、分区或数据文件），请使用以下命令：

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

对数据库启用和禁用缓存

适用于 Oracle 的 SmartIO 插件可用于对数据库启用或禁用缓存。

您可以使用此操作对备份或数据仓储 ETL（解压缩、转换和加载）操作等数据库作业临时禁用缓存。作业完成后，即可启用缓存。在数据库处于联机状态时，可以启用和禁用缓存。

对数据库启用缓存

- ◆ 请使用以下命令：

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

例如：

```
$ 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
```

对数据库禁用缓存

- ◆ 请使用以下命令：

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

例如：

```
$ 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
```

列出数据库缓存策略详细信息

使用此过程列出指定数据库对象（包括数据文件、分区或表空间）的缓存策略。

列出数据库缓存策略

- ◆ 请使用以下命令：

```
# sfcache app [cachearea=cachearea_name] oracle -S $ORACLE_SID \
-H $ORACLE_HOME -o list {--datafile=name \
|--partition=name | --tablespace=name}
```

例如，列出表空间 `stok_0` 的缓存策略：

```
$ 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
/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

列出数据库缓存统计数据

DBA 可以监控数据库级别的缓存使用率和命中率。您可以使用该统计数据评估当前缓存策略。如果命中率不符合要求，请参考 Oracle AWR 报告并更改策略。

列出数据库缓存统计数据

- ◆ 要查看数据库所有装入点的缓存统计数据, 请使用以下命令:

```
$ 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
```

关于 VxVM 卷上的数据库的 SmartIO 缓存

SmartIO 提供了应用程序模板, 以优化运行在 VxVM 卷上的数据库的缓存。SmartIO 使用这些模板, 对此数据库中特殊类型的卷应用策略。例如, 索引卷可能有与数据卷不同的缓存策略。

SmartIO 为 VxVM 卷提供下列应用程序模板:

- 模板名: **oracle**
优化运行在 VxVM 卷上的 Oracle 数据库的缓存。
- 模板名: **sybase**
优化运行在 VxVM 卷上的 Sybase 数据库的缓存。

对 VxVM 卷应用 SmartIO 数据库缓存模板

SmartIO 提供了应用程序模板, 以优化运行在 VxVM 卷上的数据库的缓存。SmartIO 使用这些模板, 对此数据库中特殊类型的卷应用策略。例如, 针对索引卷和数据卷的缓存策略。

对 VxVM 卷应用 SmartIO Sybase 缓存模板

- 1 以 root 用户身份登录。
- 2 导出 SYBASE 环境变量。

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

其中，/sybase 是 Sybase 主目录。

- 3 使用下列命令对 VxVM 卷应用 SmartIO Sybase 缓存模板：

```
# sfcache app cachearea=cachearea_name sybase \  
{olap|oltp} db_username db_server [db_name]
```

其中：

olap 或 oltp 表示应用程序负载类型。OLAP，即“联机分析处理”应用程序处理用于多维分析查询的工作负载。OLTP，即“联机事务处理”应用程序处理面向事务的工作负载，例如数据录入和检索事务处理。

db_user_name 表示数据库用户名。

db_server 表示数据库服务器名称。

db_name 表示数据库名称。数据库名称可选。如果未指定数据库名称，则会针对用户 db_user_name 下的所有数据库运行命令。

例如：

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

- 4 在出现提示时输入数据库用户的密码。
- 5 必要时可以查看缓存日志文件中的命令活动。

该日志文件是 /etc/vx/log/sfcache.log。

例如：

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

对 VxVM 卷应用 SmartIO Oracle 缓存模板

- 1 以 root 用户身份登录。
- 2 使用下列命令对 VxVM 卷应用 SmartIO Oracle 缓存模板：

```
# sfcache app cachearea=cachearea_name oracle \  
{olap|oltp} oracle_user_name ORACLE_HOME \  
ORACLE_SID [ASM_HOME [ASM_SID]]
```

其中：

`olap` 或 `oltp` 表示应用程序负载类型。OLAP，即“联机分析处理”应用程序处理用于多维分析查询的工作负载。OLTP，即“联机事务处理”应用程序处理面向事务的工作负载，例如数据录入和检索事务处理。

`oracle_user_name` 表示 Oracle 用户的用户名。

`ORACLE_HOME` 表示安装 Oracle 软件的目录。通常，`ORACLE_HOME` 值存储在 Oracle 用户的 `bash.rc` 文件或 `profile` 文件中。

`ORACLE_SID` 表示用于唯一标识数据库的系统 ID。

`ASM_HOME` 表示安装 ASM 软件的目录。

`ASM_SID` 表示 ASM 实例的系统 ID。默认情况下，该值为 `+ASM`。对于多个 ASM 实例，该值可能不同。

以下示例显示了 Oracle 数据库的 `app` 命令的变体。

第一个示例显示了直接在 VxVM 卷上创建的 Oracle 数据库。在这种情况下，`ASM_HOME` 和 `ASM_SID` 值不适用。

原始卷上的 Oracle 示例：

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

下一个示例显示了在 VxVM 卷上创建的 Oracle ASM 数据库。在这种情况下，您必须指定 `ASM_HOME`。如果需要，请指定 `ASM_SID`。

Oracle ASM 示例：

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

- 3 在出现提示时输入数据库用户的密码。
- 4 必要时可以查看缓存日志文件中的命令活动。

该日志文件是 `/etc/vx/log/sfcache.log`。

原始卷上 Oracle 日志文件示例：

```
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 日志文件示例：

```
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'
```

技术预览：Veritas InfoScale 存储环境中的分布式 SmartIO

分布式 SmartIO 是在 Veritas InfoScale 中以技术预览形式提供的新功能，用于在非生产环境中进行配置和测试。它主要针对 Oracle RAC 或 ODM。

注意：VxFS 文件系统上运行的应用程序的 SmartIO 读取缓存所支持的所有配置在分布式 SmartIO 中也受支持。

随着硬件技术的发展（例如 Infiniband 等网络互联），在存储环境中使用网络而非磁盘（作为数据共享介质）来访问和共享数据已被证明更加快速且更具成本效益。在群集中的几个节点上，可使用成本较高的 SSD 存储来更快速地缓存数据。可根据需要使用高速网络互联在群集中的任何节点上获取数据。

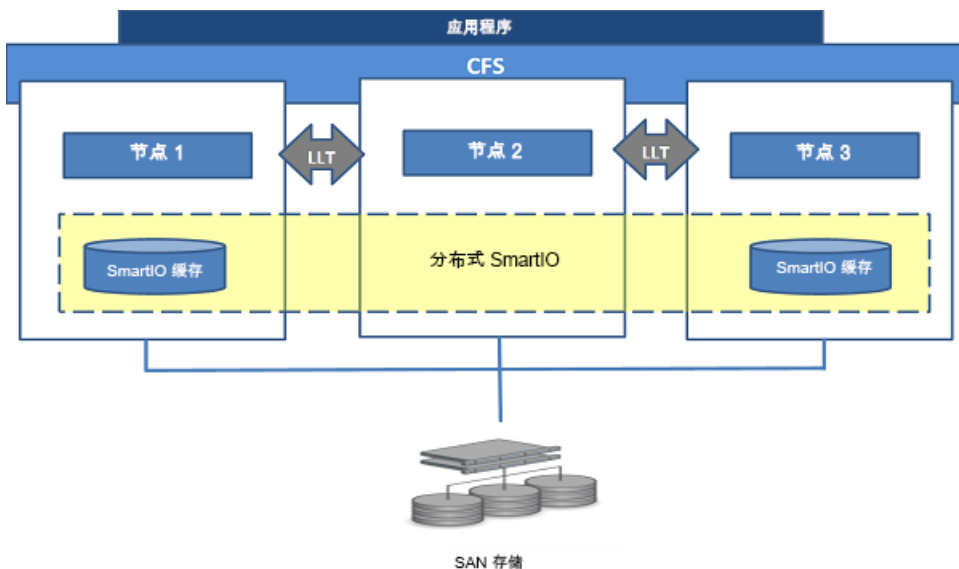
由于这些优势，Veritas InfoScale 提供了一个强大的解决方案，即“**分布式 SmartIO**”，它允许您在群集中的所有节点之间共享 SSD 资源，以便缓存频繁读取的数据。

分布式 SmartIO 功能使用 SSD 资源在本地和远程进行缓存，以提高在群集中运行的应用程序的性能。通过使用分布式 SmartIO，您可以在群集中的所有节点上使用 SSD 创建一致的缓存区域，而不管每个节点是否具有独立的本地缓存。

在分布式 SmartIO 功能中，群集中的几个或所有节点都已连接到 SmartIO 缓存。群集中的每个节点都通过高速网络互联（例如 Infiniband 或 10G 以太网）连接。将在群集中所有节点内的 SmartIO 缓存之间分布和共享读取操作 (RO) 缓存数据以及关于缓存数据的元数据。这些元数据还驻留在后端存储设备中。

请参见[分布式 SmartIO 中的读取操作 \(RO\)](#)

请参见[分布式 SmartIO 中的写入操作](#)

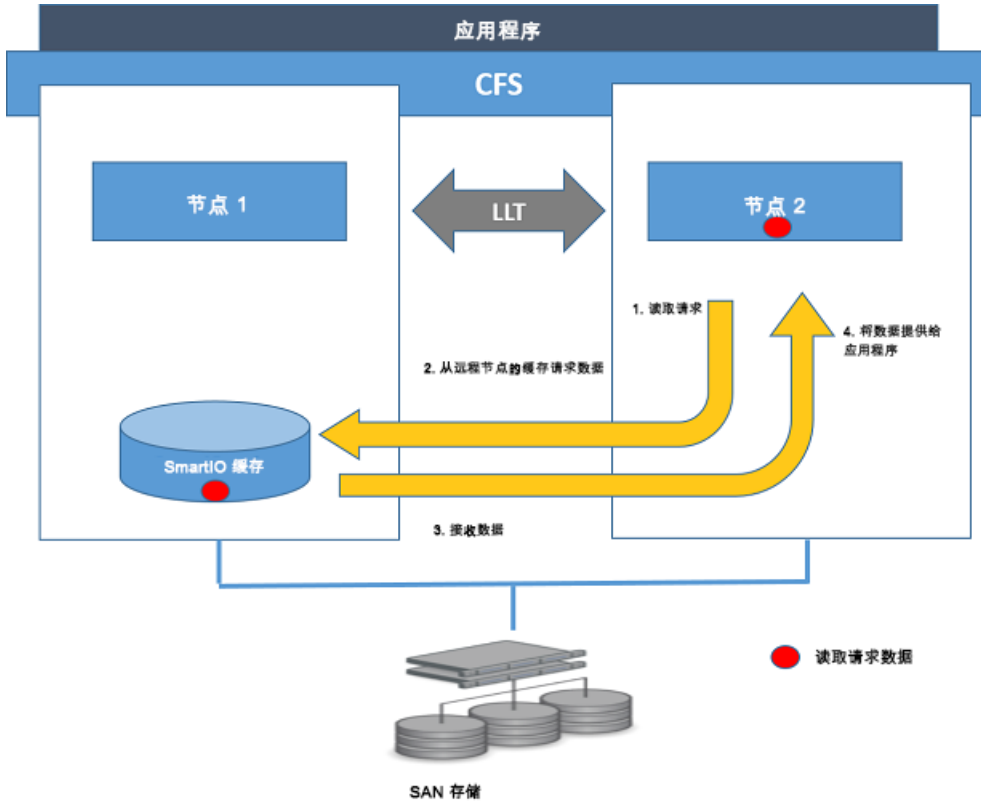


分布式 SmartIO 中的读取操作 (RO)

在群集环境中，当应用程序使用直接 I/O 执行 RO 时，分布式 SmartIO 将检查是否可以从现有节点上的本地缓存执行 RO。当现有节点上的本地缓存中没有所请求的数据时，分布式 SmartIO 会尝试从远程节点的本地缓存执行 RO。如果所请求的数据不存于任何缓存中，则会在分布式 SmartIO 中从后端存储异步呈现数据。

注意：即使群集中的一个或多个节点未连接到独立的 SmartIO（本地缓存），也可以使用群集中所有节点上的 SSD 创建一致的缓存区域。

下图概述了分布式 SmartIO 中读取操作的处理方式：

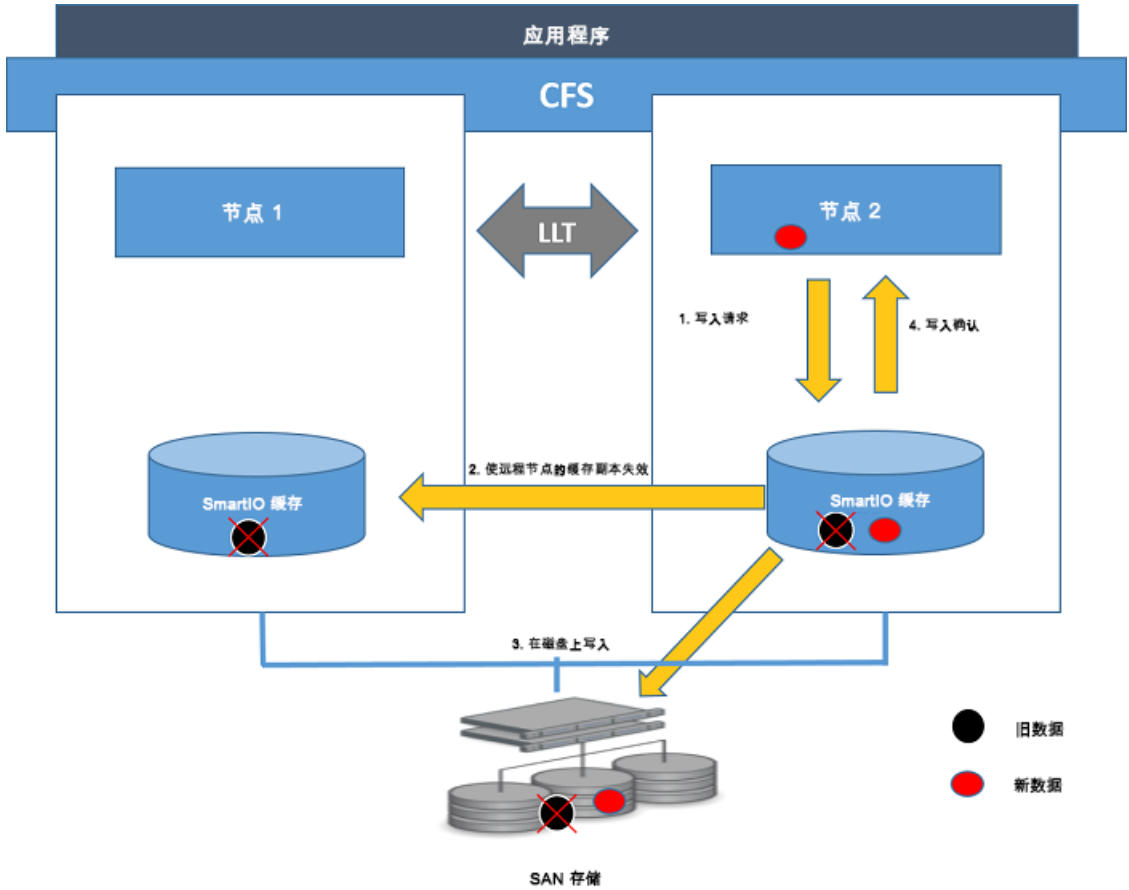


分布式 SmartIO 中的写入操作

在群集环境中，当应用程序在群集内的任何节点上执行写入操作时，它会使所有远程缓存中的数据副本失效，但执行写入操作的节点的本地缓存中的数据除外。将用新数据更新此节点的本地缓存。

注意：即使群集中的一个或多个节点未连接到独立的 SmartIO（本地缓存），也可以使用群集中所有节点上的 SSD 创建一致的缓存区域。

下图概述了分布式 SmartIO 中写入操作的处理方式：



在 Veritas InfoScale 存储环境中启用分布式 SmartIO 功能

在 Veritas InfoScale 存储环境中启用分布式 SmartIO 功能

- 1 使用 `sfcache` 命令在群集中的所需节点上创建 SmartIO 缓存。

请参见第 58 页的“创建缓存区域”。获取更多详细信息。

有关更多详细信息，请参见 `sfcache(1M)` 手册页。

- 2 装入 VxFS 文件系统时，可使用 `cluster,smartiomode=cfusion` 选项启用分布式 SmartIO。

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

- 3 要查看有关缓存使用情况的统计数据，请使用以下命令：

```
# sfcache stat
```

```
Cache Name: cache_dg1/cachevol
Cache Size: 24.97 GB
Cache Utilization: 70.8 MB ( 0.28 %)
File Systems Using Cache: 6
Writeback Cache Use Limit: Unlimited
Writeback Flush Timelag: 10 s
```

Read Cache			Writeback			
Hit Ratio	Data Read	Data Written	Hit Ratio	Data Written	rdcachename	wbcachename
Total:						
45.40 %	4.99 MB	2 MB	0.00 %	0 KB		
/mnt1:						
74.98 %	2.996 MB	1 MB	0.00 %	0 KB	cache_dg1/cachevol	-
Remote Read:						
66.62 %	1.996 MB					

输出在特定装入点的“远程读取”部分下显示“数据读取”和“命中率”统计信息。它提供了有关由 SmartIO 缓存（本地缓存）提供的数据的统计信息，以用于远程节点上的读取操作。

分布式 SmartIO 的限制

对分布式 SmartIO 的支持受以下约束的限制：

- 只有在 VxFS 文件系统上运行的应用程序才支持分布式 SmartIO 功能
- 分布式 SmartIO 功能仅在与直接 I/O 一起使用时才起作用

管理 SmartIO

本章节包括下列主题：

- [创建缓存区域](#)
- [显示有关缓存区域的信息](#)
- [启用或禁用对数据对象的缓存](#)
- [将设备添加到缓存区域](#)
- [暂停从卷缓存到缓存区域](#)
- [从缓存区域中删除设备](#)
- [销毁缓存区](#)
- [设置 VxVM 缓存区域的属性](#)
- [设置或更改 VxFS 缓存区域的缓存模式](#)
- [刷新写回缓存区域中的脏数据](#)
- [调整写回缓存](#)
- [查看 SmartIO 缓存统计数据](#)

创建缓存区域

SmartIO 引入了缓存区的概念。缓存区是 SmartIO 用来存储缓存数据及其相关元数据的存储空间。您可以创建缓存区来使用 I/O 缓存。通常，您可以将 SSD 设备或其他速度较快的设备用于缓存区。缓存区可用于 VxFS 缓存或 VxVM 缓存。您可以在每个系统上为 VxFS 缓存创建多个缓存区域，在每个系统上为 VxVM 缓存创建一个缓存区域。

要在设备上创建缓存区，请指定设备名称（磁盘访问名称）或设备上的磁盘组和卷的名称。

默认情况下，缓存区的关联类型为 **auto**。同一类型（VxVM 或 VxFS）的所有数据对象都与 **auto** 缓存区隐式关联。如果关联类型为 **noauto**，您必须将数据对象与缓存区显式关联。

在群集中，每个节点都具有独立的本地缓存区域。

创建 VxVM 缓存区

- ◆ 使用以下命令之一在 SSD 设备上创建 VxVM 缓存区：
 - 使用设备的磁盘访问名称 (*daname*) 指定一个或多个设备。设备应进行初始化，以供 VxVM 使用，并且其格式为 `cdsdisk`。

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

其中：

daname 指定要在其上创建缓存区域的设备的磁盘访问名称。

cachearea_name 指定缓存区域的自定义名称。如果未指定，SmartIO 功能会自动为缓存区域生成一个名称。

size 指定缓存区域的大小。默认情况下，缓存区会使用设备上的所有可用空间。

`--noauto` | `--auto` 指定缓存区的类型。默认值为 `--auto`。

`--nostripe|ncols=n` 指定缓存区域的布局选项。默认情况下，如果指定了两个或多个磁盘，则在条带卷上创建缓存区域。使用 `ncols=n` 选项指定条带卷的列数。使用 `--nostripe` 选项在指定磁盘的连续卷上创建缓存区域。

cache_line_size 指定 SmartIO 用于缓存的单位。当应用程序 I/O 访问数据时，SmartIO 会根据缓存块大小将数据移到缓存。您通常不需要更改 *cache_line_size*。

例如：

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

如果指定多个磁盘，缓存区域默认跨指定的磁盘进行条带化。例如：

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

- 或者，您也可以指定要用于缓存区的磁盘组和卷的名称。在这种情况下，您不能指定大小，因此此时缓存区使用整个卷。

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

其中：

dg/vol 指定要用于缓存区域的磁盘组名称和卷名称。SmartIO 将此名称用于缓存区。

--noauto | **--auto** 指定缓存区的关联类型。默认值为 **--auto**。

例如：

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

创建 VxFS 缓存区

- ◆ 使用以下命令之一在 SSD 设备上创建 VxFS 缓存区域。
 - 使用设备的磁盘访问名称 (**daname**) 指定一个或多个设备。设备应进行初始化，以供 VxVM 使用，并且其格式为 **cdsdisk**。

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

其中：

daname 指定要在其上创建缓存区域的设备的磁盘访问名称。

cachearea_name 指定缓存区域的自定义名称。如果未指定，SmartIO 功能会自动为缓存区域生成一个名称。

size 指定缓存区域的大小。默认情况下，缓存区会使用设备上的所有可用空间。

--noauto | **--auto** 指定缓存区的类型。默认值为 **--auto**。

例如：

```
# sfcache create ssd0_0
```

--nostripe|ncols=*n* 指定缓存区域的布局选项。默认情况下，如果指定了两个或多个磁盘，则在条带卷上创建缓存区域。使用 **ncols=*n*** 选项指定条带卷的列数。使用 **--nostripe** 选项在指定磁盘的连续卷上创建缓存区域。

- 或者，您也可以指定要用于缓存区的磁盘组和卷的名称。在这种情况下，您不能指定大小，因此此时缓存区使用整个卷。

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

其中：

dg/vol 指定要用于缓存区域的磁盘组名称和卷名称。SmartIO 将此名称用于缓存区。

--noauto | **--auto** 指定缓存区的类型。默认值为 **--auto**。

例如：

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

显示有关缓存区域的信息

SmartIO 使用 SSD 设备上的缓存区域进行缓存。每个缓存区域包含缓存类型、大小、关联类型、缓存状态以及用于缓存的设备等属性。缓存区域还具有关联的数据对象。

对于 VxFS 缓存区域，`sfcache list` 命令显示文件或目录的缓存模式。如果没有显示设置模式，文件或目录会继承装入点的缓存模式。如果显示设置了文件或目录的模式，会在重新装入后保留该值。显示的缓存模式可能与为装入点启用的模式有所不同。除非文件系统在 `writeback` 模式下装入，否则不会启用 `writeback` 模式。如果文件或目录设置为 `writeback` 模式，但文件系统在其他模式下装入，则文件或目录会继承装入点的缓存模式。

显示有关缓存区域的信息

- 1 使用以下命令可以显示有关系统上的缓存区的信息。

```
# sfcache list
```

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

2 使用以下命令可以显示有关某个特定缓存区的信息。

VxVM 缓存区域的示例:

```
# 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 缓存区域的示例:

```
# sfcache list sfcachearea_2
Cachearea: sfcachearea_2
Assoc Type: AUTO
Type: VxFS
Size: 7.00g
State: ONLINE
Layout: CONCAT
FStype: default
Number of Columns: 0
```

FSUUID	SIZE	MODE	MOUNTPPOINT	CACHENAME
a6178a5604a50200577a...	759.0 MB	read	/ora_inst1	sfcachearea_2
0d929856a7d60e003d38...	512.0 MB	writeback	/pdb2	sfcachearea_2
90929856b68800008141...	50.1 MB	read	/fast_recovery	sfcachearea_2
02188a569d7e0700567d...	4 KB	nocache	-	sfcachearea_2
31188a5679a90900987d...	4 KB	nocache	-	sfcachearea_2
a55b8a56db160a00fa61...	4 KB	nocache	-	sfcachearea_2

启用或禁用对数据对象的缓存

使用 SmartIO 功能时，您可以禁用或启用对某个卷或装入点的缓存。

如果配置了自动缓存，则不需要显式启用对任何数据对象的缓存。SmartIO 功能会将类型相同的所有数据对象缓存为缓存区（VxVM 或 VxFS）。默认情况下，数据对象的缓存状态为 `auto`。如果需要，可以禁用对某个特定数据对象的缓存。如果系统上存在一个 `auto` 缓存区，SmartIO 功能会缓存状态为 `auto` 或 `enabled` 的所有对象。

如果未配置自动缓存，则必须对希望 SmartIO 功能缓存的任何数据对象显式启用缓存。如果系统上存在一个 `no-auto` 缓存区，SmartIO 功能仅会缓存状态为 `enabled` 的对象。此时 SmartIO 功能不会缓存状态为 `auto`（这是默认缓存状态）的数据对象。

请参见第 64 页的“启用或禁用对文件系统的缓存”。

请参见第 65 页的“启用或禁用对数据卷的缓存”。

启用或禁用对文件系统的缓存

SmartIO 功能可让您禁用或启用对某个特定文件系统的缓存。对于群集文件系统，必须对群集中的每个节点运行命令。

当缓存区域在脱机后联机或缩小缓存区域时，启用或禁用对文件系统的缓存不是持久性的。在这种情况下，对文件系统的缓存取决于装入点上或使用 `sfcache set mode` 命令设置的缓存模式。要保持缓存状态不变，请使用 `sfcache set mode` 命令。

请参见第 68 页的“设置或更改 VxFS 缓存区域的缓存模式”。

启用对文件系统的缓存

- ◆ 要启用对文件系统的缓存，请使用以下命令。

```
# sfcache enable mount_point
```

其中：

`mount_point` 是文件系统的装入点。

禁用对文件系统的缓存

- ◆ 要禁用对文件系统的缓存，请使用以下命令。使用 `-o purge` 选项将现有缓存数据从缓存中清除掉。默认情况下，数据不会被清除。

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

其中：

`mount_point` 是文件系统的装入点。

清除文件系统的缓存数据

- ◆ 要清除文件系统的缓存数据，请使用以下命令。使用 `purge` 选项将现有缓存数据从缓存中清除掉。

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

其中：

`mount_point` 是文件系统的装入点。

`fsuuid` 指定文件系统的 UUID。指定 `fsuuid` 以清除未装入的文件系统的缓存数据。

启用或禁用对数据卷的缓存

SmartIO 可让您禁用或启用对某个特定数据卷的缓存。

启用对某个特定数据卷的缓存

- ◆ 要启用对某个特定数据卷的缓存，请使用以下命令。

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

禁用对某个特定数据卷的缓存

- ◆ 要禁用对某个特定数据卷的缓存，请使用以下命令。

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

将设备添加到缓存区域

增加缓存区域的大小时，可以将设备添加到缓存区域。此操作可以联机执行。

调整缓存区的大小

- ◆ 要调整缓存区的大小，请使用以下命令。按磁盘访问名称指定新设备，以将缓存区扩展到该设备。`maxsize` 选项可将缓存区域的大小增加到缓存区域中的设备支持的最大大小。

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

暂停从卷缓存到缓存区域

对于 VxVM 类型的缓存区域，您可以暂时停止缓存到特定卷的缓存区域，但不删除卷和缓存之间的关联。可以暂停缓存，以后再继续缓存。暂停缓存时，读取操作不会缓存任何新数据。如果已缓存的区域发生了写入，则会更新该缓存的内容。

暂停缓存

- ◆ 要暂停，请使用以下命令。

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

继续缓存

- ◆ 要继续卷的缓存，请使用以下命令。

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

从缓存区域中删除设备

如果缓存区域跨越多个设备，则可以从缓存区域中删除其中一个设备。

从缓存区域中删除设备

- 1 确保删除设备后缓存适用于剩余设备。如果需要，将缓存的大小调整为小于剩余设备大小。

如果缩小 VxFS 缓存区域，所有缓存的数据（包括统计数据）可能会被逐出。

```
# sfcache resize newsize cachearea_name
```

例如，假设当前缓存区域 `sfcachearea_1` 的最大大小为 5G。您要删除 2G 的 SSD `ssd0_0`。首先，将缓存区域大小调整为 2.5G：

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

- 2 要从缓存区域中删除设备，请使用磁盘访问名称指定要从缓存区域中删除的设备。

```
# sfcache rmdev [dname ...]
```

例如，要删除 SSD `ssd0_0`：

```
# sfcache rmdev ssd0_0
```

- 3 您可以使用 `maxsize` 选项将缓存区域的大小调整为适用于缓存区域中剩余设备的最大大小。

```
# sfcache resize maxsize cachearea_name
```

例如：

```
# sfcache resize maxsize sfcachearea_1
```

销毁缓存区

您可以销毁缓存区，而此操作会删除缓存中的所有数据。销毁缓存区之前，必须使缓存区脱机，以确保当前未执行缓存操作。

销毁缓存区

- 1 使缓存区脱机：

```
# sfcache offline cachearea_name
```

- 2 要销毁缓存区，请使用以下命令。

```
# sfcache delete cachearea_name
```

设置 VxVM 缓存区域的属性

对于 VxVM 缓存区域，可以设置下列属性：

- `memsz`。SmartIO 解决方案保存 VxVM 缓存相关元数据所需的系统 RAM 量。如果创建了 VxVM 缓存区域，SmartIO 会计算最佳可能值。您无需更改此值，除非系统 RAM 不足。
- 关联类型。关联类型为 `--auto` 或 `--noauto`。关联类型决定是否为系统上的卷自动启用缓存。
请参见第 13 页的“针对 VxVM 卷的自动缓存”。

设置内存大小

- ◆ 要设置内存大小，请使用下列命令。

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

其中：

`dg/vol` 指定用于缓存区域的磁盘组和卷。

`cachearea_name` 指定缓存区域的名称

`size` 指定核内元数据的内存的最大大小。

例如：

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

设置关联类型

- ◆ 要设置关联类型，请使用下列命令。

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

其中：

`dg/vol` 指定用于缓存区域的磁盘组和卷。

`cachearea_name` 指定缓存区域的名称

例如：

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

设置或更改 VxFS 缓存区域的缓存模式

对于 VxFS 缓存区，缓存模式决定了为指定装入点执行的缓存类型。缓存模式可以是 `nocache`、`read` 或 `writeback`。默认模式为 `read`。

VxVM 缓存区仅支持 read 模式。

缓存区的类型（VxVM 或 VxFS）无法更改。必须销毁缓存区，然后创建所需类型的新缓存区。

可以通过 `-o smartiomode` 选项设置 VxFS 装入点的缓存模式。通过 `mount` 命令设置的缓存模式代表可为装入点上的对象启用的最高级别缓存。如果指定 `nocache` 模式，装入点会禁用 SmartIO 缓存。此时无法为该装入点中的任何数据对象启用 SmartIO 缓存。要启用缓存，必须重新装入文件系统。

同样，如果在装入期间指定了读取模式，则无法为该装入点中的任何数据对象启用 SmartIO `writeback` 缓存。

设置 VxFS 装入点的缓存模式

- ◆ 要在装入 VxFS 文件系统时设置缓存模式，请使用以下命令：

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

其中：

其中 *mode* 是下列各项之一：

- writeback
- read
- nocache

更改 VxFS 装入点的缓存模式

- ◆ 要更改文件或目录的缓存模式，请使用以下命令。无法将缓存模式更改为较之 `mount_point` 设置的模式更高级别的缓存。例如，如果装入文件系统时将模式指定为 `nocache`，则无法启用读取缓存。

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

使用 `-r` 选项可使更改递归。

刷新写回缓存区域中的脏数据

使用 SmartIO，在正常操作期间，缓存中的脏数据将自动刷新到磁盘。在未装入文件系统时或在其他需要冻结文件系统的操作期间，将刷新脏数据。同时也会按固定间隔刷新脏数据。可以通过配置可调参数来控制间隔。

请参见第 70 页的“调整写回缓存”。

禁用文件的 `writeback` 缓存也会刷新该文件的任何写回脏数据。

在某些情况下，您可能希望手动触发将脏数据从缓存刷新到磁盘的操作。例如，要确保数据的一致性，应在创建阵列级别快照前刷新缓存。

您可以使用以下命令手动触发刷新脏数据的操作。

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

使用 `-r` 选项进行递归选择。

调整写回缓存

启用 `writeback` 缓存时，从磁盘读取的任何数据都会缓存，除非文件显式标记为“不进行缓存”或缓存已满。对于写入操作而言，某些写入操作会导致数据缓存。您可以加载文件，以加快应用程序运行速度。将文件固定在缓存中可确保数据不会被逐出。如果某些数据已经缓存，而磁盘的该部分内容遭到覆盖，此时 SmartIO 也会将新数据写入到缓存设备，以确保缓存数据保持最新。

如果使用数据库模板，SmartIO 会根据模板规则进行缓存。

请参见第 41 页的[“关于 VxFS 文件系统上的 Oracle 数据库的 SmartIO 缓存”](#)。

可以使用以下可调参数来调整缓存的大小，以及数据在缓存中保留的期限。

[设置每个节点用于脏数据的最大空间](#)

[设置脏数据的最长保留时间](#)

设置每个节点用于脏数据的最大空间

启用 `writeback` 时，可以配置用于脏数据的缓存量。`writeback_size` 属性设置用于每个文件系统写回数据的最大缓存区域空间数量。根据节点设置最大空间。默认情况下，不设置最大空间。如果您配置最大空间，值必须至少为 512 MB。

对于群集文件系统，`writeback` 模式中的 SmartIO 会将每个节点的缓存数据反映或镜像到其他节点的 SSD 缓存。实际的磁盘空间使用量是独立文件系统使用量的两倍。但是，最大大小中不考虑反映的数据。例如，如果 `writeback_size` 设置为 512 MB，群集文件系统最多在每个节点上使用 512 MB，总计 1024 MB。

运行以下命令以配置最大空间。对于群集文件系统，请在群集的每个节点运行该命令，以使设置应用到整个群集。

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

例如：

```
# sfcache set writeback_size=1g
```

使用以下命令可以查看当前值：

```
# sfcache stat cachearea_name
```

请参见第 71 页的[“查看 SmartIO 缓存统计数据”](#)。

设置脏数据的最长保留时间

脏数据是指缓存中尚未刷新到磁盘的数据，因此它与数据磁盘不同步。保留时间决定了脏数据可保持不刷新的期限。默认值为 10 秒。

对于群集文件系统，请在群集的每个节点运行该命令，以使设置应用到整个群集。

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

例如：

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

使用以下命令可以查看当前值：

```
# sfcache stat cachearea_name
```

请参见第 71 页的“[查看 SmartIO 缓存统计数据](#)”。

查看 SmartIO 缓存统计数据

使用 `sfcache stat` 命令可以显示系统的缓存统计数据。

[表 3-1](#) 介绍了输出中的缓存统计数据。

表 3-1 缓存统计数据

字段	说明
HIT RATIO (VxVM cache)	从缓存得到满足的总 I/O 百分比。显示读写操作的数据。
ART(Hit)ms (VxVM cache)	从缓存得到满足的 I/O 的平均响应时间。显示读取 (RD) 和写入 (WR) 的数据。
ART(Miss)ms (VxVM cache)	未从缓存得到满足的 I/O 的平均响应时间。显示读取 (RD) 和写入 (WR) 的数据。
BYTES (VxVM cache)	读取 (RD) 和写入 (WR) I/O 的总大小。
NAME (VxVM cache)	缓存区的名称。
TYPE (VxVM cache)	缓存区为 VxVM 还是 VxFS。
%CACHE (VxVM cache)	当前用于数据的缓存区的百分比。显示所有数据对象的数据。
Cache Size (VxFS cache)	缓存区的大小。
Cache Utilization (VxFS cache)	当前用于数据的缓存区的百分比。

字段	说明
File Systems Using Cache (VxFS cache)	使用缓存的文件系统数量。
Writeback Cache Use Limit (VxFS cache)	用于写回的缓存区域大小。 可以使用 <code>writeback_size</code> 属性设置大小。 如果未设置大小，字段将显示“无限”。 请参见第 70 页的 “调整写回缓存” 。
Writeback Flush Timelag	数据写到缓存与数据刷新到磁盘之间的时间间隔。如果 <code>Writeback Flush Timelag</code> 较小（例如 10 秒钟）， <code>sfcache</code> 统计数据不会显示此数据。此时数据会以更快的速度刷新到磁盘。在这种情况下，您可以根据 <code>WB Hit Ratio</code> 确定缓存使用情况。
Hit Ratio (VxFS cache)	从缓存得到满足的总 I/O 百分比。显示读写操作的数据。
Data Read (VxFS cache)	从缓存读取的数据。
Data Written (VxFS cache)	写入到缓存的数据。
Files Cached (VxFS cache)	缓存中存在的文件数量。
Files Pinned (VxFS cache)	固定在缓冲中的文件数量。
Data Pinned (VxFS cache)	固定在缓存中的数据量。

查看缓存统计数据

◆ 请使用以下命令：

```
# 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

```

查看 VxVM 缓存区的详细缓存统计数据

查看 VxVM 缓存区的详细缓存统计数据

◆ 请使用以下命令：

```
# 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

```

查看 VxFS 缓存区域的详细缓存统计数据

对于 VxFS 缓存区域，卸载和装入文件系统后，统计数据不会更改。对于群集文件系统，统计数据并不会在您重新启动群集节点后发生变更。

查看 VxFS 缓存区域的详细缓存统计数据

1 请使用以下命令：

```
# sfcache stat sfcachearea_1
```

```
TYPE: VxFS
NAME: sfcachearea_1
    Cache Name: sfcachearea_1
        Cache Size:      5 GB
        Cache Utilization: 1.426 GB (28.51 %)
File Systems Using Cache: 2
Writeback Cache Use Limit: Unlimited
    Writeback Flush Timelag: 10 s

Read Cache                                Writeback
Hit Ratio  Data Read Data Written Hit Ratio  Data Written  rdcachename  wbcachename

Total:
    7.98 %  157.7 MB  1.545 GB    0.00 %    0 KB

/pdb1:
    7.98 %  157.7 MB  1.545 GB    0.00 %    0 KB          sfcachearea_1    -

/pdb2:
    0.00 %  0 KB      0 KB        0.00 %    0 KB          sfcachearea_1  sfcachearea_2
```

2 要查看详细信息，请使用 -l 选项：

```
# sfcache stat -l
```

```
TYPE: VxFS :
```

```
    Cache Name: sfcachearea_2
```

```
        Cache Size:      7 GB
```

```
        Cache Utilization: 1.26 GB (18.00 %)
```

```
File Systems Using Cache:      6
```

```
Writeback Cache Use Limit: Unlimited
```

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

```
    Cache Name: sfcachearea_1
```

```
        Cache Size:      5 GB
```

```
        Cache Utilization: 1.447 GB (28.94 %)
```

```
File Systems Using Cache:      2
```

```
Writeback Cache Use Limit: Unlimited
```

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

```
.  
. .  
. .
```

故障排除和错误处理

本章节包括下列主题：

- [对持续或暖 VxVM 缓存的支持](#)
- [磁盘发生故障后缓存区域丢失 \(3158482\)](#)
- [重新启动后缓存未联机](#)
- [在节点出现故障后恢复写回缓存](#)

对持续或暖 VxVM 缓存的支持

暖缓存表示缓存内容在计划内重新启动期间持续保留。默认情况下，SmartIO 不为 VxVM 缓存提供暖缓存功能。缓存区域元数据在系统关闭期间不会刷新。缓存在系统重新启动后失效，无论何时都需重新启动卷。此行为称为冷缓存。

Veritas 建议您不要配置暖缓存，因为这样可能导致数据不一致。不过，在某些情况下，如果操作得当，配置暖缓存会大有益处。例如，在受控的计划内重新启动期间，您可能希望明确启用暖缓存。此功能允许您将元数据刷新到缓存中，从而创建暖缓存。

启用持续或暖缓存时，如果在缓存脱机时更新数据卷，VxVM 将检测到持续缓存并使其无效。在群集环境中，如果某个节点上的缓存区域在该节点的计划关闭期间持续存在，并在其他节点上更新了数据卷，则暖缓存内容将被视为失效，并将逐出持续缓存的数据。

对计划内重新启动启用暖缓存

- 1 在系统重新启动之前，关闭所有应用程序。
- 2 通过运行以下命令创建暖缓存：

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

如果需要，可以将此命令添加到关闭脚本中。

必须运行上述命令，然后才能重新启动应用程序或重新启动系统。否则，缓存中的数据将会清除，而且重新启动之后不会将数据填入缓存。

过时缓存的主卷故障可能导致数据损坏

如果需要还原主卷中的数据，启用暖缓存可能导致过时缓存的使用。某些情况下，过时缓存会导致数据损坏。这种情况并不常见，但如果使用下述还原方法则可能会出现：

- 从阵列快照还原
- 使用 NetBackup 或其他备份软件从备份还原主 LUN。
- VxVM 配置备份和还原。

不支持在 HA 故障转移期间迁移缓存

即使启用了暖缓存，SmartIO 也不会提供在故障转移期间迁移高可用性群集中缓存的功能。发生故障转移后，应用程序无法使用暖缓存。

磁盘发生故障后缓存区域丢失 (3158482)

SmartIO 支持多个 VxFS 缓存区域和一个 VxVM 缓存区域。如果您创建一个缓存区域且磁盘发生故障，则该缓存区域将禁用。如果在启用缓存磁盘组之前，尝试创建另一个其他类型的缓存区域，则第一个缓存区域会丢失。该缓存区域无法联机。

例如，首先创建一个 VxFS 缓存区域。磁盘发生故障，该缓存区域被禁用。现在，创建 VxVM 缓存区域。创建 VxVM 缓存区域时，SmartIO 会搜索现有的默认缓存区域。由于磁盘发生故障，因此无法找到现有的缓存区域。于是，SmartIO 便创建一个同名的 VxVM 缓存区域。现在，即使包含 VxFS 缓存区域的磁盘正常工作，SmartIO 也不会访问原始缓存区域。这种情况下，VxFS 缓存区域丢失。在此情况下丢失缓存区域不会导致任何数据丢失或数据不一致问题。

解决方法：

创建一个新的 VxFS 缓存区域。

重新启动后缓存未联机

通常，SmartIO 缓存会在系统重新启动后自动联机。

如果在重新启动后 SSD 驱动程序模块未自动加载，则需要加载驱动程序并手动将缓存磁盘组联机。

在重新启动后使缓存联机

- 1 使用 `insmod` 命令加载 SSD 驱动程序模块。

有关详细信息，请参见 Linux 文档。

- 2 执行 OS 设备的扫描：

```
# vxdisk scandisks
```

- 3 手动使缓存联机：

```
# vxdg import cachedg
```

在节点出现故障后恢复写回缓存

在 `writeback` 模式下，在将数据写入磁盘前，SmartIO 会将数据存储于缓存中。如果包含缓存数据的设备出现故障，则在系统重新进入联机状态后，必须将缓存区域中未刷新的数据刷新到磁盘。在刷新缓存数据之前，在 `writeback` 模式下缓存的文件数据可能不会完整地呈现在磁盘上。

在某些情况下，包含脏写回数据的 SmartIO 缓存无法刷新。例如，包含缓存区域的 SSD 设备可能出现了错误。当缓存设备重新进入联机状态后，将跳过刷新这些文件的脏数据。无法访问具有暂停脏数据的文件。这些文件（已删除的文件除外）上的任何 I/O 都将返回 I/O 错误 (EIO)。

系统会在 `syslog` 中显示以下错误消息：

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

如果要还原对这些文件的访问，请使用以下过程。

还原对不可访问文件的访问

- 1 要还原对文件的访问，请使用以下命令。此命令需要 root 权限。

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

例如：

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

对于群集文件系统，在群集的每个节点上运行 `sfcache restore-access` 命令。

- 2 还原对文件或目录的访问后，还原对装入点的访问。

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

例如：

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

- 3 要启用文件系统的写回缓存，请使用以下命令：

```
# sfcache disable /testFS
```

```
# sfcache enable /testFS
```

命令参考资料

本附录包括下列主题：

- [SmartIO 命令参考](#)

SmartIO 命令参考

表 A-1 列出了用于 SmartIO 功能的命令。

请参见 `sfcache(1M)` 手册页。

表 A-1 SmartIO 命令参考

命令	说明
<code>sfcache app</code>	应用指定的模板名称。
<code>sfcache create</code>	创建缓存区。
<code>sfcache delete</code>	删除指定的缓存区。
<code>sfcache disable</code>	禁用对指定数据对象的缓存。
<code>sfcache enable</code>	启用对指定数据对象的缓存。
<code>sfcache flush</code>	刷新此文件系统或缓存的任何写回数据。
<code>sfcache list</code>	显示已缓存的文件系统或卷及其缓存使用情况。
<code>sfcache load</code>	将指定文件加载到缓存区。
<code>sfcache maxsize</code>	显示已置备用于缓存的设备中可用空间的数量。
<code>sfcache offline</code>	禁止 VxFS 或 VxVM 使用缓存区。

命令	说明
<code>sfcache online</code>	将缓存区显式指定为可用。
<code>sfcache pin</code>	将文件或目录标记为保留在缓存中，直至该文件或目录被删除、裁截或取消固定。
<code>sfcache purge</code>	删除指定文件系统已缓存的内容。
<code>sfcache resize</code>	调整指定缓存区的大小。
<code>sfcache restore-access</code>	启用对丢失写回数据的文件的读或写访问权限。此命令无法还原丢失的数据。
<code>sfcache rmdev</code>	将一个或多个设备取消用于缓存。
<code>sfcache set</code>	设置指定属性的值。
<code>sfcache stat</code>	显示缓存统计数据，包括缓存命中率、未命中次数和平均读写延迟。
<code>sfcache unpin</code>	使文件或目录脱离固定状态。

索引

C

- 查看卷缓存统计数据 73
- 查看默认缓存统计数据 71
- 查看文件系统缓存统计数据 74
- 查看详细缓存统计数据 73–74
- 从缓存中取消固定文件 22

D

- 调整缓存区域的大小 65–66
- 读取缓存 17
- 读取缓存模式 68
- 多个缓存区域
 - 验证 39

F

- 非缓存缓存模式 68

G

- 更改缓存模式 68
- 固态驱动器 (SSD)
 - 关于 6
- 管理缓存
 - 销毁缓存区 67

H

- 缓存磁盘组 11
- 缓存模式
 - 设置 68
- 缓存区域
 - 创建 58
 - 监视 14, 21, 30, 39
 - 删除设备 66
 - 设置属性 68
 - 添加设备 65
 - 验证 14, 21, 30
- 缓存使用率统计数据
 - 查看 71
- 缓存行为
 - 自定义 22
- 恢复缓存 66

J

- 减小缓存区域 66
- 将文件固定到缓存中 22
- 将文件加载到缓存中 22

S

- sfcache stat 命令 71
- SmartIO
 - 关于 6
- 删除缓存区 67
- 删除设备 66
- 设置缓存模式 68
- 设置缓存属性 68
- 数据对象 64
 - 禁用缓存 64
 - 启用缓存 64
- 数据卷
 - 禁用缓存 65
 - 启用缓存 65

T

- 添加设备 65
- 统计数据
 - 查看缓存使用率 71

V

- Veritas File System (VxFS)
 - 读取缓存 17
 - 配置读取缓存 18
 - 写回缓存 26–27
 - 写入缓存 28
 - 自定义缓存 22
- Veritas Volume Manager (VxVM)
 - SSD 上的缓存 13
 - 读取缓存 11
 - 自动缓存 13

W

- 文件系统
 - 禁用缓存 64

启用缓存 64

X

销毁缓存区 67

写回缓存 26

写回缓存模式 68

写入缓存 28

Z

在 SSD 设备上缓存 VxVM 13

在读取模式下缓存 VxFS 18

在写入模式下缓存 VxFS 28

暂停缓存 66

增加缓存区域的大小 65

自动缓存 13