



FlexCache回写

ONTAP 9

NetApp
August 12, 2024

目录

FlexCache回写	1
FlexCache回写概述	1
FlexCache回写架构	2
FlexCache回写用例	5
FlexCache回写前提条件	6
FlexCache回写互操作性	6
启用并管理FlexCache回写	7

FlexCache回写

FlexCache回写概述

FlexCache回写是在ONTAP 9.15.1中推出的一种备用操作模式、用于在缓存中写入数据。通过回写、可以将写入提交到缓存中的稳定存储、并向客户端确认、而无需等待数据传输到源站。数据会异步转储回源站。因此、可以构建一个全球分布式文件系统、使特定工作负载和环境能够以接近本地的速度执行写入、从而显著提高性能。



ONTAP 9.12.1引入了一项回写功能、可作为公共预览。这称为回写版本1 (wbv1)、不应视为与ONTAP 9.15.1中的回写相同、后者称为回写版本2 (wbv2)。



从2024年7月开始、以前以PDF格式发布的技术报告中的内容已与ONTAP产品文档集成。ONTAP安全文档现在包含TR-4743: 《ONTAP 9.11.1中的FlexCache》中的内容。

回写与绕写

自FlexCache在ONTAP 9.5中推出以来、它一直是一种可读写缓存；但是、它在绕写模式下运行。缓存中的写入操作会发送到初始卷、以提交到稳定存储。初始服务器成功将写入提交到稳定存储后、它会确认写入缓存。然后、缓存会向客户端确认写入。这样、每次写入都会导致在缓存和源站之间遍历网络。FlexCache回写将更改此设置。



升级到ONTAP 9.15.1后、您可以将传统的绕写缓存转换为回写缓存、并在必要时返回到绕写缓存。但是、如果出现问题、这可能会使读取诊断日志变得更加困难。

	绕写	回写
ONTAP 版本	9.6多个	9.15.1+
用例	读取繁重的工作负载	写入量大的工作负载
数据提交时间	来源	缓存
客户体验	类似于WAN	LAN-like
限制	每个原产地100个	每个原产地10个
"Cap Theorem"	对分区可用并容错	可用且一致

FlexCache回写术语

了解使用FlexCache回写的关键概念和术语。

期限	定义
dirty 数据	已提交到缓存中的稳定存储、但尚未转储到源站的数据。
专用锁定委派(XLD)	按文件授予缓存的协议级锁定权限。此授权允许缓存向客户端分发专用写入锁定、而无需联系源站。

期限	定义
共享锁定委派(SLD)	按文件授予缓存的协议级锁定权限。此授权允许缓存向客户端分发共享读取锁定、而无需联系源站。
回写	一种FlexCache操作模式、其中、对缓存的写入会提交到该缓存中的稳定存储、并立即向客户端确认。数据会异步写入源站。
绕写式	一种FlexCache操作模式、其中、对缓存的写入会转发到初始卷以提交到稳定存储。提交后、初始服务器将确认对缓存的写入、而缓存将确认对客户端的写入。
脏数据记录系统(DDRS)	一种专有机制、可在启用了回写功能的缓存中按文件跟踪脏数据。
原产地	包含所有FlexCache缓存卷的源数据的FlexGroup或FlexVol。它是唯一的数据来源、可协调锁定、并确保100%的数据一致性、货币性和一致性。
缓存	FlexGroup是FlexCache初始卷的稀疏缓存卷。

一致、当前和一致

FlexCache是NetApp的解决方案、可随时随地提供正确的数据。FlexCache始终保持100%一致、当前和一致：

- *一致：*无论访问数据的位置、数据都是相同的。
- *current：*数据始终是最新的。
- *一致：*数据正确/未损坏。

FlexCache回写架构

FlexCache绕过在设计时充分考虑了强大的一致性。传统的绕写操作模式和ONTAP 9.15.1中引入的新回写操作模式均可确保访问的数据始终保持100%一致、最新且一致。

以下概念详细介绍了FlexCache回写的操作方式。

委派

锁定委派和数据委派有助于FlexCache保持回写和绕写缓存数据的一致性、一致性和最新性。源服务器会编排这两个委派。

锁定委派

锁定委派是指初始服务器按文件授予缓存的协议级锁定授权、用于根据需要向客户端发出协议锁定。其中包括[专用锁定委派\(XD\)](#)和[共享锁定委派\(SLD\)](#)。

XLD和回写

为了确保ONTAP不必协调冲突写入、将向客户端请求写入文件的缓存授予XLD。重要的是、任何文件在任何时候只能存在一个XLD、这意味着一个文件一次不会有多个写入程序。

当写入文件的请求进入启用了回写的缓存时、将执行以下步骤：

1. 缓存会检查是否已为请求的文件创建了XLD。如果是这样、只要另一个客户端未向缓存中的文件写入数据、

它就会为客户端授予写入锁定。如果缓存没有用于请求文件的XLD、则它将从原始位置请求一个。这是一个经过集群间网络的专有调用。

2. 从缓存收到XLD-请求后、初始服务器将检查另一个缓存中的文件是否存在未完成的XLD-。如果是这样、它将调用该文件的XLD、从而触发从该缓存转储到原始缓存的任何 **脏数据**。
3. 一旦将该缓存中的脏数据转储回来并提交到源站的稳定存储、源站将为请求缓存授予文件的XLD。
4. 收到文件的XLD后、缓存会将锁定授予客户端、写入将开始。

序列图中介绍了涵盖其中某些步骤的高级序列图 [\[write-back-sequence-diagram\]](#)。

从客户端的角度来看、所有锁定都将像写入标准FlexVol或FlexGroup一样工作、在请求写入锁定时可能会有一个小延迟。

在当前迭代中、如果启用了回写的缓存保存了文件的XLD、则ONTAP将阻止其他缓存对该文件的*任何*访问、包括 READ 操作。



每个原始成分卷的XLD限制为170。

数据委派

数据委派是指初始卷为缓存提供的每个文件的保证、保证为该文件缓存的数据是最新的。只要缓存具有文件的数据委派、它就可以向客户端提供该文件的缓存数据、而无需联系源站。如果缓存没有文件的数据委派、它必须联系源站以接收客户端请求的数据。

在回写模式下、如果在另一个缓存或原始位置为文件创建了XLD、则文件的数据委派将被撤消。这样可以有效地将文件与所有其他缓存中的客户端和源服务器隔开、即使是读取也是如此。这是一种权衡、必须确保旧数据永远不会被访问。

启用了回写的缓存上的读取操作通常类似于绕写缓存上的读取操作。在启用了绕写和回写的缓存中、如果请求的文件在启用了回写的缓存(而不是发出读取的缓存)具有排他写锁定、则初始性能可能会 READ 受到影响。必须撤消XLD、并且必须先将脏数据提交到源站、然后才能处理其他缓存上的读取。

跟踪脏数据

从缓存回写源站的操作是异步进行的。这意味着脏数据不会立即写入源站。ONTAP使用脏数据记录系统来跟踪每个文件的脏数据。每个脏数据记录(脏数据记录、DDR)代表特定文件的大约20 MB脏数据。当正在写入文件时、ONTAP将在填充两个DDRR并写入第三个DDR后开始将脏数据转储回。这样、写入期间缓存中会保留大约40 MB的脏数据。对于有状态协议(NFSv4.x、SMB)、在关闭文件后、剩余的40 MB数据将转储回原始卷。对于无状态协议(NFSv3)、当在其他缓存上请求访问该文件时、或者在该文件闲置两分钟或更长时间(最长为五分钟)后、将转储40 MB的数据。有关计时器触发或空间触发的异常数据转储的详细信息,请参见 [\[缓存擦除器\]](#)。

除了CDR和擦除器之外、某些前端NAS操作还会触发文件的所有脏数据的转储:

- SETATTR
 - SETATTRs: 在缓存中仅 mtime`处理、`atime`或`ctime`修改。
- CLOSE
- OPEN 在另一个缓存中
- READ 在另一个缓存中

- REaddir 在另一个缓存中
- REaddirPLUS 在另一个缓存中
- WRITE 在另一个缓存中

已断开连接模式

如果文件的XLD保存在绕写缓存中、并且该缓存与原始缓存断开连接、则仍允许在其他缓存和原始缓存中读取该文件。如果XLD由启用了回写的缓存保留、则此行为会有所不同。在这种情况下、如果缓存断开连接、则对文件的读取将在任何地方挂起。这有助于确保保持100%的一致性、一致性和一致性。允许在绕写模式下进行读取、因为保证源站拥有已向客户端写入确认的所有可用数据。在断开连接期间的回写模式下、初始卷无法保证已启用回写的缓存中写入并确认的所有数据都在断开连接之前写入到初始卷。

如果文件的具有XLD的缓存长时间断开连接、系统管理员可以手动在原始位置撤消XLD。这样、该文件的IO便可在运行正常的缓存和原始卷上恢复。



手动恢复XLD将导致断开连接的缓存中文件的任何脏数据丢失。只有在缓存和源站之间发生灾难性中断时、才应手动删除XLD。

缓存擦除器

ONTAP中存在一些擦除了程序、它们会根据特定事件运行、例如计时器即将过期或违反空间阈值。擦除程序会对正在擦除的文件进行独占锁定、从而有效冻结该文件的IO、直到擦除完成为止。

擦尘器包括：

- 缓存上的*基于时间的清理程序：*此清理程序每五分钟启动一次，并擦除任何未经修改的文件两分钟。如果此文件的任何脏数据仍在缓存中、则此文件的IO将被静置并触发回写。回写完成后、IO将恢复。
- 与缓存中基于mtime的清理程序非常相似，该程序也每五分钟运行一次。但是、它会将未修改的任何文件进行15分钟的反转、从而重新调用索引节点的委派。此擦除程序不会启动任何回写。
- **RW**基于限制的清理程序在源站： ONTAP可监控每个源站成分卷分发的RW锁定委派的数量。如果此数字超过170、则ONTAP将开始擦除最近最少使用的写入锁定委派(Least Recently Used、LRU)。
- *缓存上基于空间的擦除程序：*如果FlexCache卷已满90%、则会擦除缓存、并按LRU逐出。
- *基于空间的擦除程序：*如果FlexCache初始卷已满90%、则会擦除缓存、并按LRU逐出。

序列图

这些顺序图显示了绕写模式与回写模式之间的写入确定差异。

绕写

[FlexCache绕写序列图] | [flexcache-write-around-sequence-diagram.png](#)

回写

[FlexCache回写顺序图] | [flexcache-write-back-sequence-diagram.png](#)

FlexCache回写用例

这些写入配置文件最适合启用了回写的FlexCache。您应测试工作负载、以确定回写或绕写是否可提供最佳性能。



回写不能替代绕写。虽然回写是针对写入量大的工作负载设计的、但对于许多工作负载来说、绕写仍然是更好的选择。

目标工作负载

文件大小

文件大小的重要性低于在和调用文件之间发出的写入次数 `OPEN CLOSE`。小型文件本身调用次数较少 `WRITE`、因此不太适合回写。在和调用之间、大型文件可能会写入更多数据 `OPEN CLOSE`、但这并不能保证。

写入大小

从客户端写入时、除了写入调用之外、还会涉及其他NAS调用：

- `CREATE`
- `OPEN`
- `CLOSE`
- `REaddir/REaddirPLUS`
- `SETATTR`: `SETATTR` 仅修改、或在缓存中处理的调用 `mtime atime ctime`。

这些调用必须在初始位置进行处理、并对在启用了回写的缓存上为正在操作的文件累积的任何脏数据触发回写操作。文件的IO将暂停、直到回写完成。

了解这些呼叫必须遍历WAN有助于确定适合回写的工作负载。通常、在和调用之间可以执行的写入越多、而不发出其他任何一个调用 `<write-size,above>`、性能增益回写效果就越 `OPEN CLOSE` 好。

写后读取

写入后读取工作负载以往在FlexCache中的性能较差。这是由于9.15.1之前的绕写操作模式造成的。`WRITE``对文件的调用必须在初始位置提交、后续 ``READ` 调用必须将数据提取回缓存。这会导致这两种操作都会对WAN造成影响。因此、对于采用绕写模式的FlexCache、不建议在写入后读取工作负载。随着9.15.1中引入回写功能、数据现在会提交到缓存中、并可立即从缓存中读取、从而消除了WAN方面的负面影响。如果您的工作负载在FlexCache卷上包括后读写、则应将缓存配置为在回写模式下运行。



如果写入后读取是工作负载的关键部分、则应将缓存配置为在回写模式下运行。

写后写入

当文件在缓存中累积脏数据时、缓存会将数据异步写入源站。这自然会导致客户端关闭文件时、脏数据仍在等待转储回源。如果刚刚关闭但仍包含脏数据的文件再次打开或写入、则写入操作将暂停、直到所有脏数据都转储到源站为止。

延迟注意事项

当FlexCache在回写模式下运行时、它对NAS客户端更有利、因为缓存与源站之间的延迟会增加。但是、在一定

程度上、回写开销会超过在低延迟环境中获得的优势。在某些NetApp测试中、回写优势始于缓存与初始服务器之间的最小延迟8毫秒。此延迟因工作负载而异、因此请务必进行测试、以了解您的优势。

下图显示了NetApp实验室测试中回写的返回点。x`轴是文件大小、`y轴是经过的时间。此测试使用NFSv3、挂载的 rsize 和 wsize 为256 KB、WAN延迟为64毫秒。执行此测试时、会对缓存和初始卷使用一个小型ONTAP Select实例、并执行单线程写入操作。结果可能有所不同。

[返回点] | [flexcache-write-back-point-of-return-nfs3.png](#)



回写不应用于集群内缓存。如果初始缓存和缓存位于同一集群中、则会发生集群内缓存。

FlexCache回写前提条件

在以回写模式部署FlexCache之前、请确保满足这些软件、许可和系统配置要求。

ONTAP 版本

- 原点*Must*运行的是ONTAP 9.15.1.或更高版本。
- 任何需要在回写模式*Must*下运行的缓存集群都运行ONTAP 9.15.1.或更高版本。
- 任何不需要在回写模式下运行的缓存集群均可运行任何受支持的ONTAP版本。

许可

您购买的ONTAP附带了FlexCache (包括回写模式操作)。不需要额外的许可证。

对等

- 初始集群和缓存集群必须为 "已建立集群对等"
- 原始集群和缓存集群上的服务器虚拟机(SVM)必须 "已为Vserver对等" 具有FlexCache选项。



您无需将缓存集群与另一个缓存集群建立对等关系。也无需将缓存SVM与另一个缓存SVM建立对等关系。

FlexCache回写互操作性

了解在回写模式下部署FlexCache时的这些互操作性注意事项。

ONTAP 版本

要使用回写操作模式、缓存和原始服务器*必须*都运行ONTAP 9.15.1.或更高版本。



不需要启用回写缓存的集群可以运行早期版本的ONTAP、但该集群只能在绕写模式下运行。

您的环境可以混合使用多种ONTAP版本。

集群	ONTAP 版本	是否支持回写?
原产地	ONTAP 9.15.1.	不适用
集群1	ONTAP 9.15.1.	是的。
集群2	ONTAP 9.14.1	否

集群	ONTAP 版本	是否支持回写?
原产地	ONTAP 9.14.1	不适用
集群1	ONTAP 9.15.1.	否
集群2	ONTAP 9.15.1.	否



在中 [\[example2-table\]](#), 两个集群都不能启用回写模式, 因为初始集群未运行ONTAP 9.151或更高版本, 这是一项严格要求。

客户端互操作性

ONTAP通常支持的任何客户端均可访问FlexCache卷、而不管该卷是在绕写模式还是回写模式下运行。有关受支持客户端的最新列表, 请参阅NetApp的 "[互操作性表](#)"。

尽管客户端版本无关紧要、但客户端必须足够新才能支持NFS3、NFSv4.0、NFSv4.1、SMB2.x或SMB3.xSMB1和NFSv2是已弃用的协议、不受支持。

回写和绕写

如中所示 [\[example1-table\]](#), 在回写模式下运行的FlexCache可以与在绕写模式下运行的缓存共存。建议将绕写与回写与特定工作负载进行比较。



如果工作负载在回写和绕写之间的性能相同、请使用绕写。

ONTAP功能互操作性

有关FlexCache功能互操作性的最新列表, 请参阅 "[FlexCache卷支持和不支持的功能](#)"。

启用并管理FlexCache回写

从ONTAP 9.15.1开始、您可以在FlexCache卷上启用FlexCache回写模式、以便为具有大量写入工作负载的边缘计算环境和缓存提供更好的性能。您还可以根据需要确定是在FlexCache卷上启用回写还是在卷上禁用回写。

在缓存卷上启用回写后、写入请求将发送到本地缓存、而不是原始卷。

开始之前

您必须处于高级权限模式。

创建启用了回写的新FlexCache卷

步骤

您可以使用ONTAP系统管理器或ONTAP命令行界面创建启用了回写的新FlexCache卷。

System Manager

1. 如果FlexCache卷与初始卷位于不同的集群上、请创建集群对等关系：
 - a. 在本地集群上，单击*保护>概述*。
 - b. 展开*Intercluster Settings*，单击*Add Network Interfaces*，然后将集群间接口添加到集群中。
在远程集群上重复此操作。
 - c. 在远程集群上、单击*保护>概述*。单击  "Cluster Peers"(集群对等方)部分，然后单击*Generate Passwords*。
 - d. 复制生成的密码短语并将其粘贴到本地集群中。
 - e. 在本地集群上的"Cluster Peers"下、单击*对等集群*并为本地集群和远程集群建立对等关系。
2. 如果FlexCache卷与原始卷位于不同的集群上、请创建SVM对等关系：
在* Storage VM对等方*下、依次单击  和*对等Storage VM*以建立Storage VM对等关系。
如果FlexCache卷位于同一集群上、则不能使用System Manager创建SVM对等关系。
3. 选择 * 存储 > 卷 *。
4. 选择 * 添加 *。
5. 选择*更多选项*，然后选择*为远程卷添加缓存*。
6. 选择*启用FlexCache回写*。

命令行界面

1. 如果要创建的 FlexCache 卷位于其他集群中，请创建集群对等关系：
 - a. 在目标集群上、创建与数据保护源集群的对等关系：

```
cluster peer create -generate-passphrase -offer-expiration  
MM/DD/YYYY HH:MM:SS|1...7days|1...168hours -peer-addr  
<peer_LIF_IPs> -initial-allowed-vserver-peers <svm_name>,...|*  
-ipSPACE <ipSPACE_name>
```

从 ONTAP 9.6 开始，默认情况下，在创建集群对等关系时会启用 TLS 加密。支持使用 TLS 加密在初始卷和 FlexCache 卷之间进行集群间通信。如果需要，您还可以为集群对等关系禁用 TLS 加密。

```
cluster02::> cluster peer create -generate-passphrase -offer
-expiration 2days -initial-allowed-vserver-peers *

                Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
                Expiration Time: 6/7/2017 08:16:10 EST
Initial Allowed Vserver Peers: *
                Intercluster LIF IP: 192.140.112.101
                Peer Cluster Name: Clus_7ShR (temporary generated)

Warning: make a note of the passphrase - it cannot be displayed
again.
```

- a. 在源集群上，将源集群身份验证到目标集群：

```
cluster peer create -peer-addr <peer_LIF_IPs> -ip-space <ip-space>
```

```
cluster01::> cluster peer create -peer-addr
192.140.112.101,192.140.112.102
```

Notice: Use a generated passphrase or choose a passphrase of 8 or more characters.

To ensure the authenticity of the peering relationship, use a phrase or sequence of characters that would be hard to guess.

```
Enter the passphrase:
Confirm the passphrase:
```

```
Clusters cluster02 and cluster01 are peered.
```

2. 如果FlexCache卷与原始卷位于不同的SVM中、请使用创建SVM对等关系 flexcache 作为应用程序：

- a. 如果SVM位于不同集群中、请为对等SVM创建SVM权限：

```
vserver peer permission create -peer-cluster <cluster_name>
-vserver <svm-name> -applications flexcache
```

以下示例说明了如何创建适用于所有本地 SVM 的 SVM 对等权限：

```
cluster1::> vserver peer permission create -peer-cluster cluster2
-vserver "*" -applications flexcache
```

Warning: This Vserver peer permission applies to all local Vservers. After that no explicit "vserver peer accept" command required for Vserver peer relationship creation request from peer cluster "cluster2" with any of the local Vservers. Do you want to continue? {y|n}: y

a. 创建SVM对等关系:

```
vserver peer create -vserver <local_SVM> -peer-vserver
<remote_SVM> -peer-cluster <cluster_name> -applications flexcache
```

3. 创建启用了回写的FlexCache卷:

```
volume flexcache create -vserver <cache_vserver_name> -volume
<cache_flexgroup_name> -aggr-list <list_of_aggregates> -origin
-volume <origin_flexgroup> -origin-vserver <origin_vserver name>
-justion-path <junction_path> -is-writeback-enabled true
```

在现有FlexCache卷上启用FlexCache回写

您可以使用ONTAP系统管理器或ONTAP命令行界面在现有FlexCache卷上启用FlexCache回写。

System Manager

1. 选择*存储>卷*并选择现有FlexCache卷。
2. 在卷的"概述"页面上，单击右上角的*Edit*。
3. 在*Edit Volumes*窗口中，选择*Enable FlexCache write-back*。

命令行界面

1. 在现有FlexCache卷上启用回写:

```
volume flexcache config modify -volume <cache_flexgroup_name> -is
-writeback-enabled true
```

检查是否已启用FlexCache回写

步骤

您可以使用System Manager或ONTAP命令行界面来确定是否已启用FlexCache回写。

System Manager

1. 选择*存储>卷*并选择一个卷。
2. 在卷*Overview*中，找到* FlexCache详细信息*，并检查FlexCache卷上的FlexCache回写是否设置为*Enabled"。

命令行界面

1. 检查是否已启用FlexCache回写：

```
volume flexcache config show -volume cache -fields is-writeback-enabled
```

在FlexCache卷上禁用回写

在删除FlexCache卷之前、您需要禁用FlexCache回写。

步骤

您可以使用System Manager或ONTAP命令行界面禁用FlexCache回写。

System Manager

1. 选择*存储>卷*并选择已启用FlexCache回写的现有FlexCache卷。
2. 在卷的"概述"页面上，单击右上角的*Edit*。
3. 在*Edit Volumes*窗口中，取消选择*Enable FlexCache write-back*。

命令行界面

1. 禁用回写：

```
volume flexcache config modify -volume <cache_vol_name> -is-writeback-enabled false
```

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。