



Microsoft SQL Server

Enterprise applications

NetApp
May 03, 2024

目录

Microsoft SQL Server	1
基于ONTAP的Microsoft SQL Server	1
数据库配置	1
存储配置	8
使用NetApp管理软件保护Microsoft SQL Server数据	19
使用ONTAP进行Microsoft SQL Server灾难恢复	20
保护ONTAP上的Microsoft SQL Server安全	21

Microsoft SQL Server

基于ONTAP的Microsoft SQL Server

ONTAP为您的Microsoft SQL Server数据库提供企业级安全性和性能解决方案、同时还提供世界一流的工具体来管理您的环境。



本文档将取代先前发布的技术报告_TR-4590: 《采用ONTAP的Microsoft SQL Server最佳实践指南》

NetApp假定读者具备以下方面的实践知识:

- ONTAP 软件
- NetApp SnapCenter作为备份软件、其中包括:
 - 适用于 Microsoft Windows 的 SnapCenter 插件
 - 适用于SQL Server的SnapCenter插件
- Microsoft SQL Server架构和管理

本最佳实践部分的范围仅限于根据NetApp为存储基础架构建议的设计原则和首选标准进行技术设计。端到端实施已超出范围。

有关NetApp产品之间的配置兼容性、请参见 "[NetApp 互操作性表工具 \(IMT\)](#)"。

Microsoft SQL Server工作负载

在部署SQL Server之前、您必须了解SQL Server实例支持的应用程序的数据库工作负载要求。每个应用程序对容量、性能和可用性有不同的要求、因此每个数据库的设计都应以最佳方式支持这些要求。许多组织使用应用程序要求定义SLA、将数据库分为多个管理层。SQL Server工作负载可以描述如下:

- OLTP数据库通常也是组织中最关键的数据库。这些数据库通常为面向客户的应用程序提供支持、并被视为公司核心运营的关键。任务关键型OLTP数据库及其支持的应用程序通常具有SLA、这些SLA要求高性能级别、并且对性能下降和可用性敏感。它们也可能是"无中断故障转移集群"或"无中断可用性"组的候选对象。这些类型数据库的I/O混合通常具有75%到90%的随机读取和25%到10%的写入特征。
- 决策支持系统(DSS)数据库也可以称为数据仓库。这些数据库在许多依靠分析来开展业务的企业中至关重要。在运行查询时、这些数据库对CPU利用率和磁盘读取操作非常敏感。在许多组织中、DSS数据库在月份、季度和年终都是最重要的此工作负载的读取I/O混合比例通常为100%。

数据库配置

Microsoft SQL Server CPU配置

要提高系统性能、您需要修改SQL Server设置和服务器配置、以便使用适当数量的处理器来执行。

超线程

超线程是英特尔专有的同步多线程(SMT)实施、它可以改进在x86微处理器上执行的并行化(多任务处理)。

使用超线程的硬件允许逻辑超线程CPU在操作系统中显示为物理CPU。然后、SQL Server会看到操作系统提供的物理CPU、并可使用超线程处理器。这样可通过提高并行处理能力来提高性能。

此处需要注意的是、每个SQL Server版本都有自己的计算能力限制。有关详细信息、请参见计算容量限制(按SQL Server版本)。

SQL Server许可有两种选择。第一种称为服务器+客户端访问许可证(CAL)模式；第二种是按处理器核心模式。尽管使用server + CAL策略可以访问SQL Server中提供的所有产品功能、但每个插槽的硬件限制为20个CPU核心。即使每个插槽的CPU核心数超过20个的服务器具有SQL Server Enterprise Edition + CAL、应用程序也无法在该实例上一次使用所有这些核心。

下图显示了启动后的SQL Server日志消息、指示核心限制的强制实施。

日志条目表示SQL Server启动后使用的核心数。

```
2017-01-11 07:16:30.71 Server      Microsoft SQL Server 2016
(RTM) - 13.0.1601.5 (X64)
Apr 29 2016 23:23:58
Copyright (c) Microsoft Corporation
Enterprise Edition (64-bit) on Windows Server 2016
Datacenter 6.3 <X64> (Build 14393: )

2017-01-11 07:16:30.71 Server      UTC adjustment: -8:00
2017-01-11 07:16:30.71 Server      (c) Microsoft Corporation.
2017-01-11 07:16:30.71 Server      All rights reserved.
2017-01-11 07:16:30.71 Server      Server process ID is 10176.
2017-01-11 07:16:30.71 Server      System Manufacturer:
'FUJITSU', System Model: 'PRIMERGY RX2540 M1'.
2017-01-11 07:16:30.71 Server      Authentication mode is MIXED.
2017-01-11 07:16:30.71 Server      Logging SQL Server messages
in file 'C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\Log\ERRORLOG'.
2017-01-11 07:16:30.71 Server      The service account is 'SEA-
TM\FUJIA2R30$'. This is an informational message; no user action
is required.
2017-01-11 07:16:30.71 Server      Registry startup parameters:
-d C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\DATA\master.mdf
-e C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\Log\ERRORLOG
-l C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\DATA\mastlog.ldf
-T 3502
-T 834
2017-01-11 07:16:30.71 Server      Command Line Startup
Parameters:
-s "MSSQLSERVER"
2017-01-11 07:16:30.72 Server      SQL Server detected 2 sockets
with 18 cores per socket and 36 logical processors per socket,
72 total logical processors; using 40 logical processors based
on SQL Server licensing. This is an informational message; no
user action is required.
2017-01-11 07:16:30.72 Server      SQL Server is starting at
```

因此、要使用所有CPU、应使用每个处理器的核心许可证。有关SQL Server许可的详细信息、请参见 ["SQL Server 2022: 您的现代数据平台"](#)。

CPU关联性

除非遇到性能问题、否则您不太可能需要更改处理器关联性默认值、但仍然值得了解它们是什么以及它们的工作原理。

SQL Server通过两个选项支持处理器关联性：

- CPU相关性掩码
- 相关性I/O掩码

SQL Server使用操作系统中可用的所有CPU (如果选择了按处理器核心许可证)。它会在所有CPU上创建计划程序、以便充分利用任何给定工作负载的资源。多任务处理时，操作系统或服务上的其他应用程序可以将进程线程从一个处理器切换到另一个处理器。SQL Server是一个资源密集型应用程序、发生这种情况时、性能可能会受到影响。为了最大限度地减少影响、您可以对处理器进行配置、使所有SQL Server负载都定向到一组预先选择的处理器。这可通过使用CPU相关性掩码来实现。

相关性I/O掩码选项可将SQL Server磁盘I/O绑定到一部分CPU。在SQL Server OLTP环境中、此扩展可以提高发出I/O操作的SQL Server线程的性能。

最大平行度(MAXDOP)

默认情况下、如果选择了按处理器核心许可证、则SQL Server会在查询执行期间使用所有可用的CPU。

尽管这对于大型查询很有用、但它可能会出现发生原因性能问题并限制并发性。一种更好的方法是、将并行处理限制为单个CPU插槽中的物理核心数。例如、如果服务器具有两个物理CPU插槽、每个插槽具有12个核心、而不考虑超线程、则MAXDOP应设置为12。MAXDOP不能限制或指定要使用的CPU。相反、它会限制单个批处理查询可使用的CPU数量。



MAXDOP NetApp建议对于数据仓库等DSS，从50开始，并在需要时探索调整或调整。请确保在进行更改时测量应用程序中的关键查询。

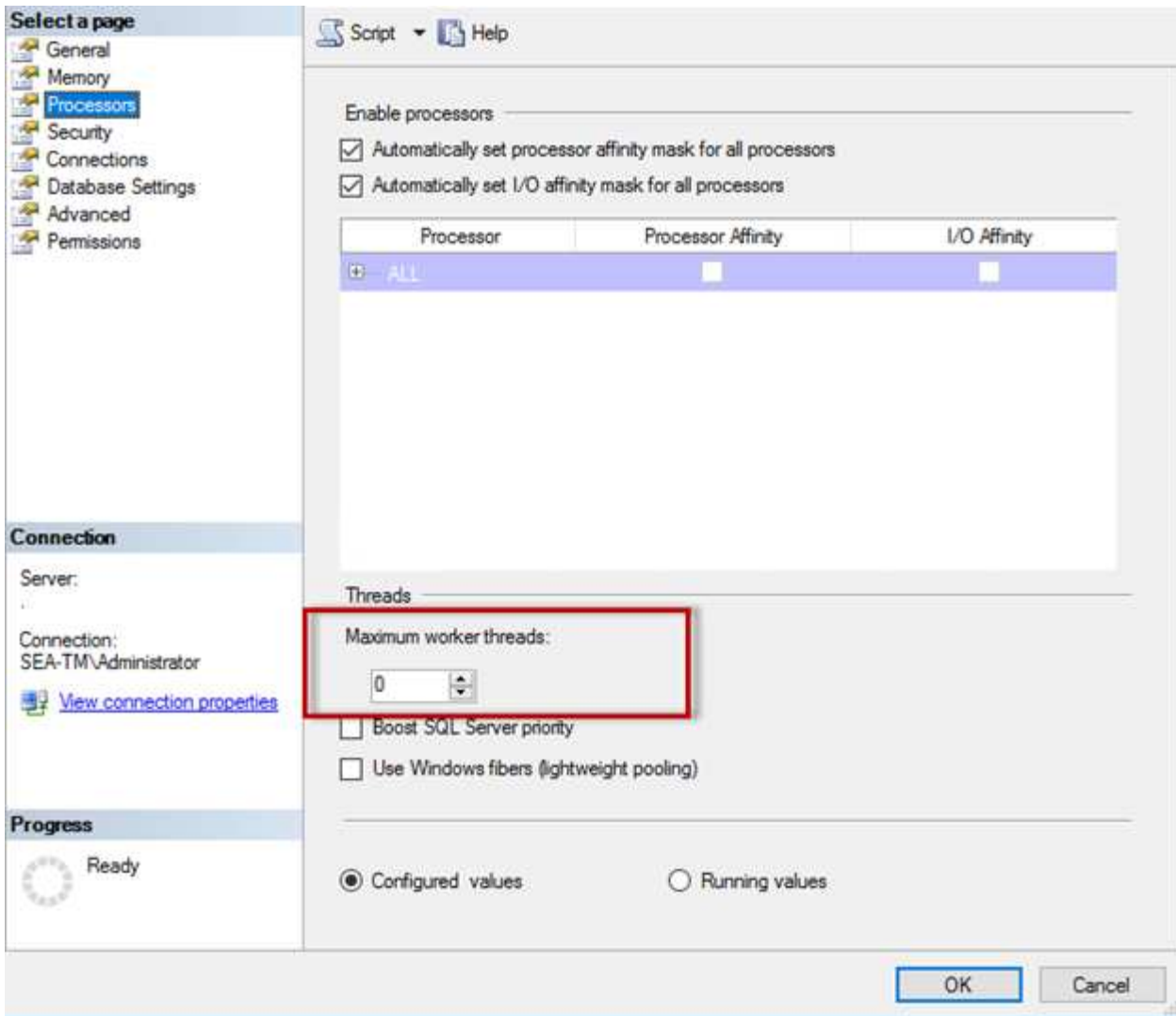
最大工作线程数

最大工作线程数选项有助于在大量客户端连接到SQL Server时优化性能。

通常、系统会为每个查询请求创建一个单独的操作系统线程。如果与SQL Server同时建立了数百个连接、则每个查询请求一个线程会占用大量系统资源。最大工作线程数选项可使SQL Server创建一个工作线程池来处理大量查询请求、从而有助于提高性能。

默认值为0、这允许SQL Server在启动时自动配置工作线程数。这适用于大多数系统。最大工作线程数是一个高级选项、没有经验丰富的数据库管理员(Database Administrator、DBA)的帮助、不得更改。

应在何时配置SQL Server以使用更多工作线程？如果每个计划程序的平均工作队列长度超过1、则向系统添加更多线程可能会更有帮助、但前提是负载不受CPU制约或遇到任何其他繁重等待。如果出现上述任一情况、添加更多线程将毫无帮助、因为它们最终会等待其他系统瓶颈。有关最大工作线程数的详细信息、请参见 ["配置最大工作线程数服务器配置选项"](#)。



使用SQL Server Management Studio配置最大工作线程数。

The following example shows how to configure the max work threads option using T-SQL.

```
EXEC sp_configure 'show advanced options', 1;  
GO  
RECONFIGURE ;  
GO  
EXEC sp_configure 'max worker threads', 900 ;  
GO  
RECONFIGURE;  
GO
```

Microsoft SQL Server内存配置

下一节将介绍如何配置SQL Server内存设置以优化数据库性能。

最大服务器内存

最大服务器内存选项用于设置SQL Server实例可以使用的最大内存量。

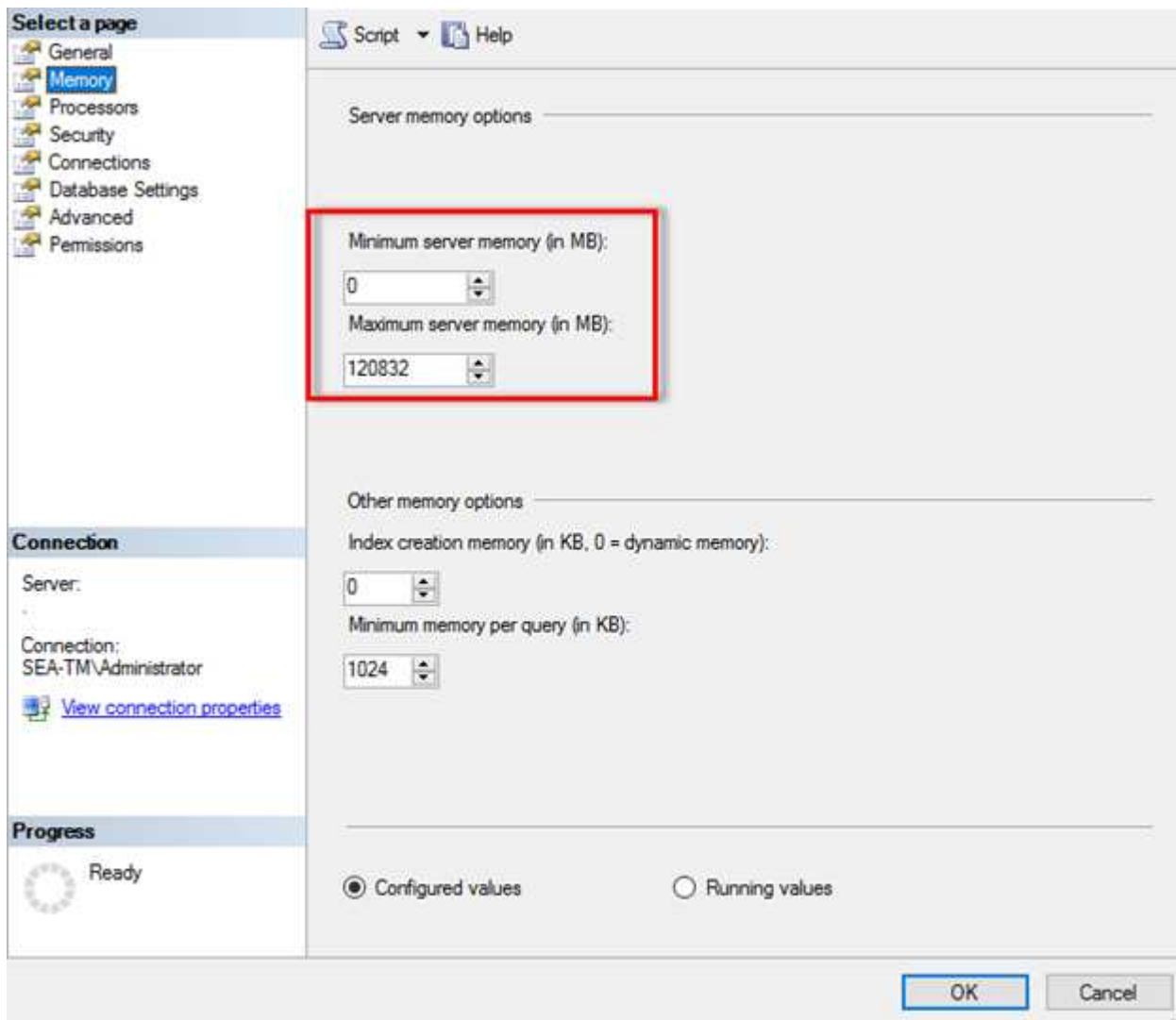
通常、如果多个应用程序运行在运行SQL Server的同一服务器上、并且您希望保证这些应用程序具有足够的内存来正常运行、则使用此方法。

某些应用程序仅在启动时使用可用的内存、即使需要也不会请求更多内存。这就是最大服务器内存设置的作用所在。

在包含多个SQL Server实例的SQL Server集群上、每个实例可能会争用资源。为每个SQL Server实例设置内存限制有助于保证每个实例的最佳性能。



*RAM NetApp建议*为操作系统至少保留4GB到6GB的内存，以避免出现性能问题。



使用SQL Server Management Studio调整最小和最大服务器内存。

使用SQL Server Management Studio调整最小或最大服务器内存需要重新启动SQL Server服务。您可以使用Transact SQL (T-SQL)通过以下代码调整服务器内存：

```
EXECUTE sp_configure 'show advanced options', 1
GO
EXECUTE sp_configure 'min server memory (MB)', 2048
GO
EXEC sp_configure 'max server memory (MB)', 120832
GO
RECONFIGURE WITH OVERRIDE
```

非一致内存访问

非一致内存访问(Nonuniform Memory Access, NMA)是一种内存访问优化方法、可帮助提高处理器速度、而不增加处理器总线上的负载。

如果在安装了SQL Server的服务器上配置了NUMA、则无需进行其他配置、因为SQL Server可以识别NUMA、并且在NUMA硬件上性能良好。

索引创建内存

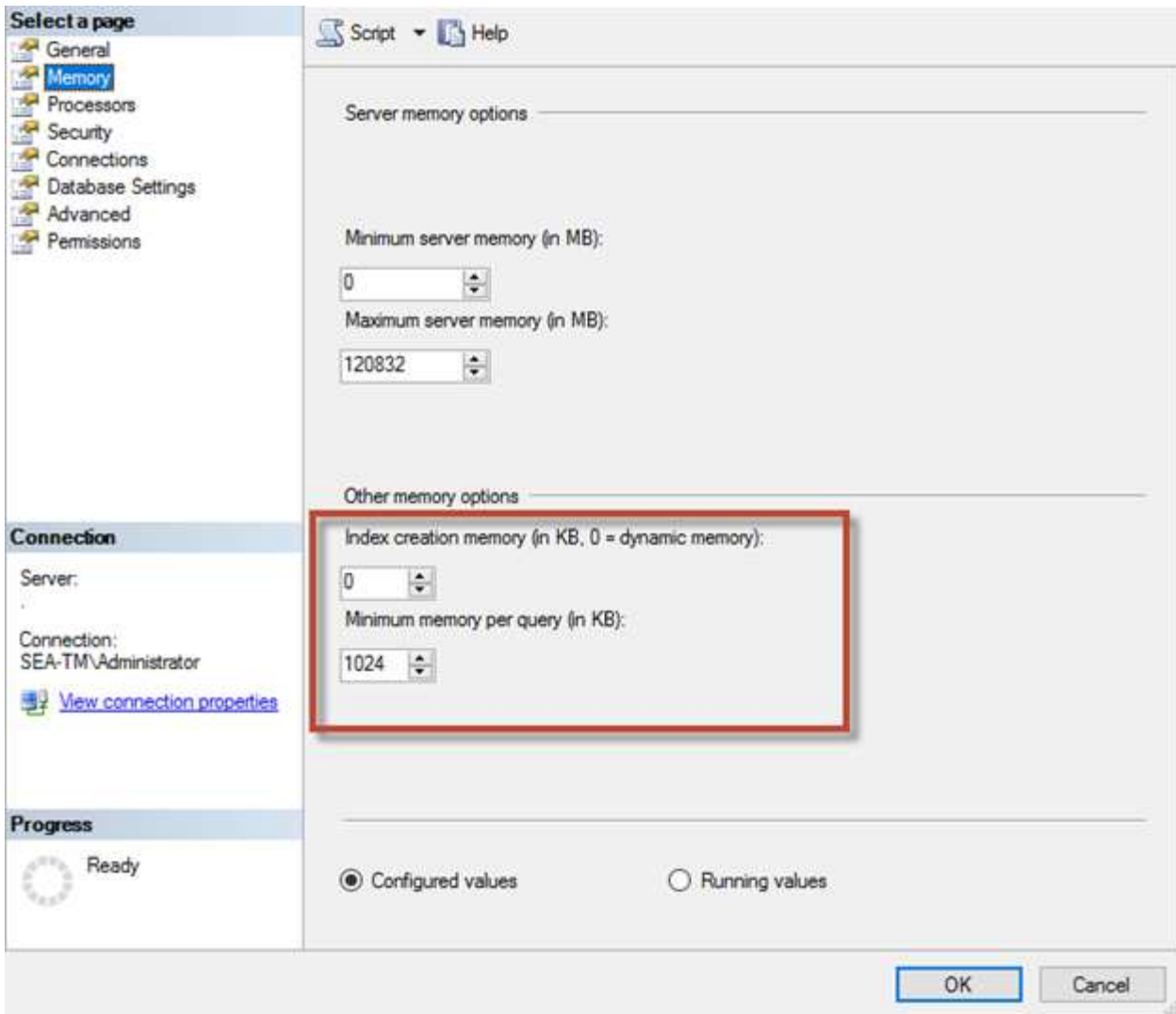
index create memory选项是另一个高级选项、通常不应更改。

它控制最初为创建索引而分配的最大RAM量。此选项的默认值为0、这意味着它由SQL Server自动管理。但是、如果在创建索引时遇到困难、请考虑增加此选项的值。

每个查询的最小内存

运行查询时、SQL Server会尝试分配最佳内存量、以便高效运行。

默认情况下、“每个查询的最小内存”设置会为要运行的每个查询分配>=到1024KB的内存。最佳做法是、将此设置保留为默认值0、以允许SQL Server动态管理为索引创建操作分配的内存量。但是、如果SQL Server的RAM超过高效运行所需的RAM、则提高此设置可以提高某些查询的性能。因此、只要SQL Server、任何其他应用程序或操作系统未使用服务器上的可用内存、则提高此设置就可以提高SQL Server的整体性能。如果没有可用内存、则增加此设置可能会影响整体性能。



缓冲池扩展

通过缓冲池扩展、可以将NVRAM扩展与数据库引擎缓冲池无缝集成、从而显著提高I/O吞吐量。

并非每个SQL Server版本都提供缓冲池扩展。它仅适用于64位SQL Server Standard、Business Intelligence和Enterprise版本。

缓冲池扩展功能可使用非易失性存储(通常为SSD)扩展缓冲池缓存。扩展后、缓冲池可以容纳更大的数据库工作集、从而强制在RAM和SSD之间分页I/O、并有效地将小型随机I/O从机械磁盘卸载到SSD。由于SSD的延迟较低且随机I/O性能较高、因此缓冲池扩展功能可显著提高I/O吞吐量。

缓冲池扩展功能具有以下优势：

- 提高了随机I/O吞吐量
- 降低I/O延迟
- 提高了事务处理吞吐量
- 利用更大的混合缓冲池提高读取性能
- 一种可利用现有和未来低成本内存的缓存架构



- NetApp建议*将缓冲池扩展配置为：
- 确保向SQL Server主机提供SSD支持的LUN (例如NetApp AFF)、以便将其用作缓冲池扩展目标磁盘。
- 扩展文件的大小必须等于或大于缓冲池。

以下示例显示了一个T-SQL命令、用于设置32 GB的缓冲池扩展名。

```
USE master
GO
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'P:\BUFFER POOL EXTENSION\SQLServerCache.BUFFER POOL
EXTENSION', SIZE = 32 GB);
GO
```

Microsoft SQL Server共享实例与专用实例

可以将多个SQL Server配置为每个服务器一个实例、也可以配置为多个实例。正确的决策通常取决于多种因素、例如服务器是用于生产还是开发、实例是否被视为对业务运营和性能目标至关重要。

共享实例配置最初可能更易于配置、但可能会导致资源被划分或锁定的问题、进而导致在共享SQL Server实例上托管数据库的其他应用程序出现性能问题。

对性能问题进行故障排除可能非常复杂、因为您必须确定哪个实例是根发生原因。此问题会与操作系统许可证和SQL Server许可证的成本进行权衡。如果应用程序性能至关重要、则强烈建议使用专用实例。

Microsoft在服务器级别为每个核心(而不是每个实例)授予SQL Server许可证。因此、为了节省许可成本、数据库管理员尝试尽可能多地安装服务器可以处理的SQL Server实例、这可能会导致以后出现重大性能问题。



*SQL NetApp建议*尽可能选择专用的SQL Server实例，以获得最佳性能。

存储配置

Microsoft SQL Server存储注意事项

通过将ONTAP存储解决方案与Microsoft SQL Server相结合、可以创建企业级数据库存储设计、以满足当今最苛刻的应用程序要求。

要优化这两种技术、了解SQL Server I/O模式和特征至关重要。SQL Server数据库的存储布局设计合理、可支持SQL Server的性能以及SQL Server基础架构的管理。良好的存储布局还可以成功完成初始部署、并随着业务的增长逐步平稳地扩展环境。

数据存储设计

对于不使用SnapCenter 执行备份的SQL Server数据库、Microsoft建议将数据和日志文件放在在不同的驱动器上。对于同时更新和请求数据的应用程序、日志文件会占用大量写入资源、而数据文件(取决于您的应用程序)会占用大量读/写资源。对于数据检索、不需要日志文件。因此、可以通过放在其自己驱动器上的数据文件来满足数据请求。

创建新数据库时、Microsoft建议为数据和日志指定单独的驱动器。要在创建数据库后移动文件、必须使数据库脱机。有关Microsoft的更多建议、请参见 "[将数据和日志文件放在不同的驱动器上](#)"。

聚合

聚合是NetApp存储配置的最低级别存储容器。互联网上存在一些旧文档、建议将IO分隔到不同的底层驱动器组上。不建议在ONTAP中使用此选项。NetApp已使用数据文件和事务日志文件分开的共享和专用聚合执行了各种I/O工作负载特征化测试。测试表明、一个具有更多RAID组和驱动器的大型聚合可以优化和提高存储性能、并使管理员更易于管理、原因有两个：

- 一个大型聚合可使所有驱动器的I/O功能对所有文件可用。
- 一个大型聚合可以最高效地利用磁盘空间。

对于高可用性(HA)、请将SQL Server Always On可用性组二级同步副本放在聚合中的一个单独Storage Virtual Machine (SVM)上。为了实现灾难恢复、请将异步副本放在灾难恢复站点中单独存储集群中的聚合上、并使用NetApp SnapMirror技术复制内容。NetApp建议在聚合中至少留出10%的可用空间、以获得最佳存储性能。

Volumes

NetApp FlexVol卷会创建并驻留在聚合中。此术语有时会引起混淆、因为ONTAP卷不是LUN。ONTAP卷是数据的管理容器。卷可以包含文件、LUN甚至S3对象。卷不会占用空间、它仅用于管理包含的数据。

卷设计注意事项

在创建数据库卷设计之前、请务必了解SQL Server I/O模式和特征如何根据工作负载以及备份和恢复要求而有所不同。请参见以下适用于灵活卷的NetApp建议：

- 避免在主机之间共享卷。例如、虽然可以在一个卷中创建2个LUN并将每个LUN共享到不同的主机、但应避免这种情况、因为它会使管理复杂化。
- 使用NTFS挂载点而不是驱动器盘符、以超过Windows中26个驱动器盘符的限制。使用卷挂载点时、一般建议为卷标签指定与挂载点相同的名称。
- 适当时、配置卷自动调整策略以帮助防止出现空间不足情况。17采用ONTAP的Microsoft SQL Server最佳实践指南©2022 NetApp、Inc. 保留所有权利。
- 如果在SMB共享上安装SQL Server、请确保在SMB/Cifs卷上启用了Unicode以创建文件夹。
- 将卷中的Snapshot预留值设置为零、以便于从操作角度进行监控。
- 禁用快照计划和保留策略。请改用SnapCenter来协调SQL Server数据卷的Snapshot副本。
- 将SQL Server系统数据库放在一个专用卷上。
- tempdb是一个系统数据库、SQL Server会使用它作为临时工作空间、尤其是用于I/O密集型DBCC CHECKDB操作。因此、请将此数据库放在具有一组单独磁盘轴的专用卷上。在卷数量是一项挑战的大型环境中、您可以在仔细规划后将tempdb整合到较少的卷中、并将其与其他系统数据库存储在同一个卷中。tempdb的数据保护不是高优先级、因为每次重新启动SQL Server时都会重新创建此数据库。

- 将用户数据文件(.mdf)放在不同的卷上、因为它们是随机读/写工作负载。通常、创建事务日志备份的频率比创建数据库备份的频率更高。因此、请将事务日志文件(.ldf)放置在与数据文件不同的卷或VMDK上、以便可以为每个卷或VMDK创建独立的备份计划。这种分离还会将日志文件的顺序写入I/O与数据文件的随机读/写I/O隔离开来、并显著提高SQL Server性能。

LUN

- 确保用户数据库文件和用于存储日志备份的日志目录位于不同的卷上、以防止保留策略在SnapVault技术中使用快照时覆盖这些快照。
- 确保SQL Server数据库与包含非数据库文件(如全文搜索相关文件)的LUN驻留在不同的LUN上。
- 将数据库二级文件(作为文件组的一部分)放在不同的卷上可提高SQL Server数据库的性能。只有当数据库的.mdf文件不与任何其他.mdf文件共享其LUN时、此分隔才有效。
- 如果使用DiskManager或其他工具创建LUN、请确保在格式化LUN时将分区的分配单元大小设置为64K。
- 请参见 "[Microsoft Windows和本机MPIO、并遵循适用于现代SAN的ONTAP最佳实践](#)" 在MPIO属性中将Windows上的多路径支持应用于iSCSI设备。

Microsoft SQL Server数据库文件和文件组

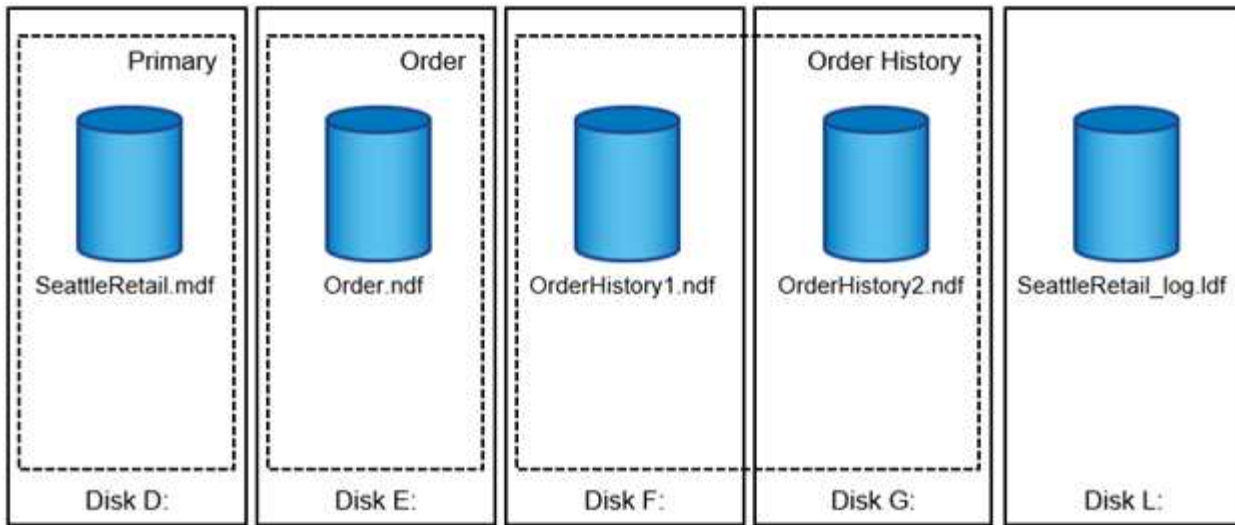
在初始部署阶段、在ONTAP上正确放置SQL Server数据库文件至关重要。这样可以确保最佳性能、空间管理、备份和还原时间、并可根据您的业务需求进行配置。

从理论上讲、SQL Server (64位)支持每个实例32、767个数据库以及524、272 TB的数据库大小、尽管典型安装通常包含多个数据库。但是、SQL Server可以处理的数据库数量取决于负载和硬件。SQL Server实例托管数十个、数百个甚至数千个小型数据库的情况并不少见。

每个数据库由一个或多个数据文件和一个或多个事务日志文件组成。事务日志存储有关数据库事务以及每个会话所做的所有数据修改的信息。每次修改数据时、SQL Server都会在事务日志中存储足够的信息、以便撤消(回滚)或重做(重放)操作。SQL Server事务日志是SQL Server在数据完整性和稳定性方面声誉的重要组成部分。事务日志对于SQL Server的运行异常、一致性、隔离和持久性(ACID)功能至关重要。一旦对数据页面进行任何更改、SQL Server就会写入事务日志。每个数据操作语言(DML)语句(例如SELECT、INSERT、UPDATE-DELETE)都是一个完整的事务、事务日志可确保执行整个基于集合的操作、从而确保事务的正常运行。

每个数据库都有一个主数据文件、默认情况下、该文件的扩展名为.mdf。此外、每个数据库都可以具有二级数据库文件。默认情况下、这些文件的扩展名为.NDF。

所有数据库文件都分组到文件组中。文件组是逻辑单元、可简化数据库管理。它们允许在逻辑对象放置和物理数据库文件之间进行隔离。创建数据库对象表时、您可以指定应将其放置在哪个文件组中、而不必担心基础数据文件配置。



通过将多个数据文件放在文件组中、您可以将负载分布在不同的存储设备上、这有助于提高系统的I/O性能。相反、事务日志不会从多个文件中受益、因为SQL Server会按顺序写入事务日志。

通过将文件组中的逻辑对象放置与物理数据库文件分隔开、您可以微调数据库文件布局、从而最大程度地利用存储子系统。例如、将产品部署到不同客户的独立软件供应商(ISV)可以在部署阶段根据底层I/O配置和预期数据量调整数据库文件数量。这些更改对于应用程序开发人员来说是透明的、他们将数据库对象放在文件组中、而不是数据库文件中。



* NetApp建议*避免将主文件组用于除系统对象之外的任何其他对象。为用户对象创建单独的文件组或一组文件组可简化数据库管理和灾难恢复、尤其是在大型数据库的情况下。

您可以在创建数据库或向现有数据库添加新文件时指定初始文件大小和自动增长参数。SQL Server在选择应将数据写入哪个数据文件时使用比例填充算法。它会将大量数据按比例写入文件中的可用空间。文件中的可用空间越多、它处理的写入就越多。



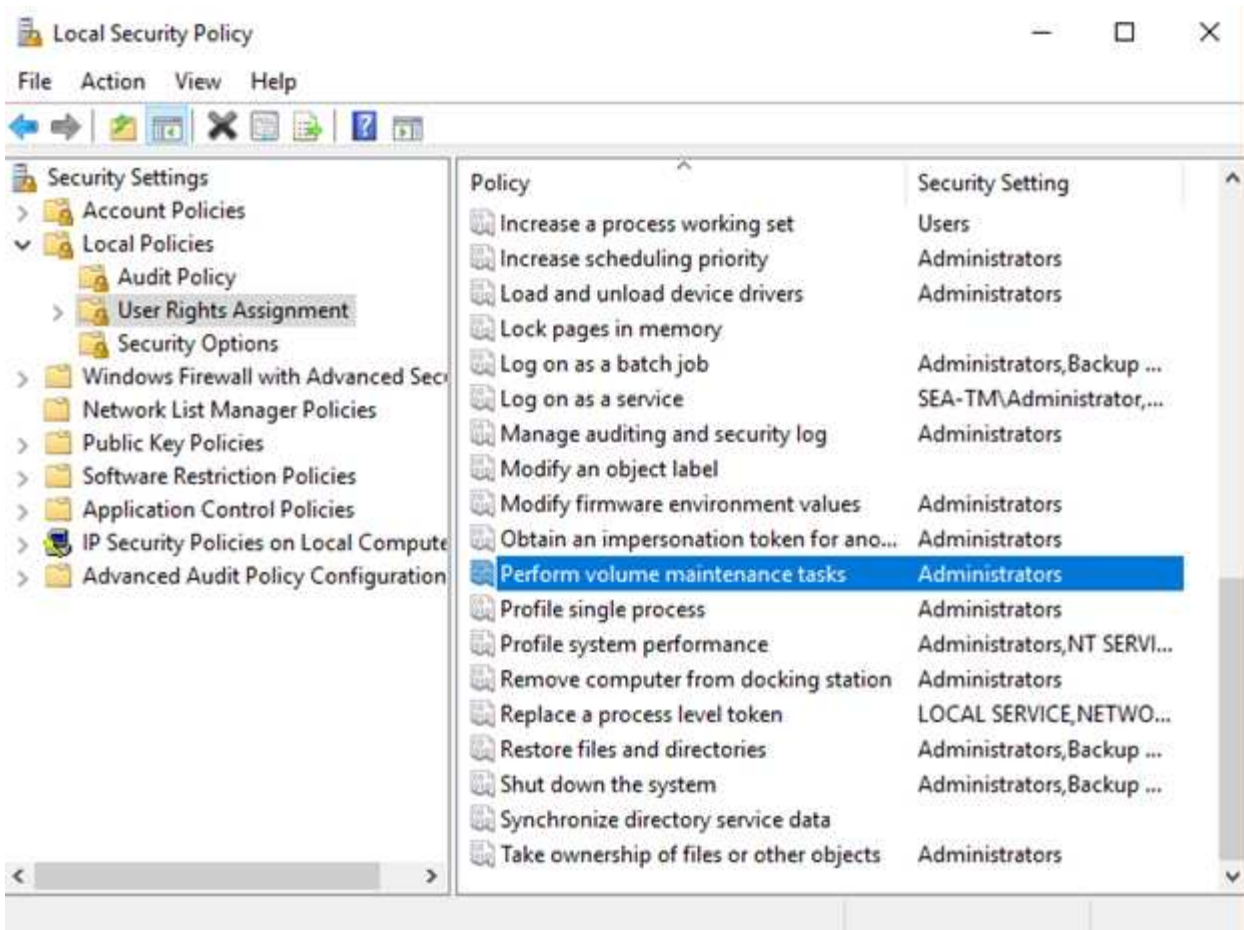
* NetApp建议*单个文件组中的所有文件都具有相同的初始大小和自动增长参数，并且增长大小以MB而不是百分比为单位定义。这有助于比例填充算法均匀地平衡数据文件之间的写入活动。

每当SQL Server增加文件大小时、它都会用零填充新分配的空间。该进程会阻止所有需要写入相应文件的会话、或者在事务日志增长时生成事务日志记录。

SQL Server始终将事务日志零、并且此行为无法更改。但是、您可以通过启用或禁用即时文件初始化来控制数据文件是否置零。启用即时文件初始化有助于加快数据文件增长速度、并减少创建或还原数据库所需的时间。

即时文件初始化会带来较小的安全风险。启用此选项后、数据文件的未分配部分可能包含先前删除的操作系统文件中的信息。数据库管理员可以检查此类数据。

您可以通过向SQL Server启动帐户添加SA_Manage_volume_name权限(也称为"执行卷维护任务")来启用即时文件初始化。您可以在本地安全策略管理应用程序(secpol.msc)下执行此操作、如下图所示。打开"Perform volume Maintenance Task (执行卷维护任务)"权限的属性、并将SQL Server启动帐户添加到其中的用户列表中。



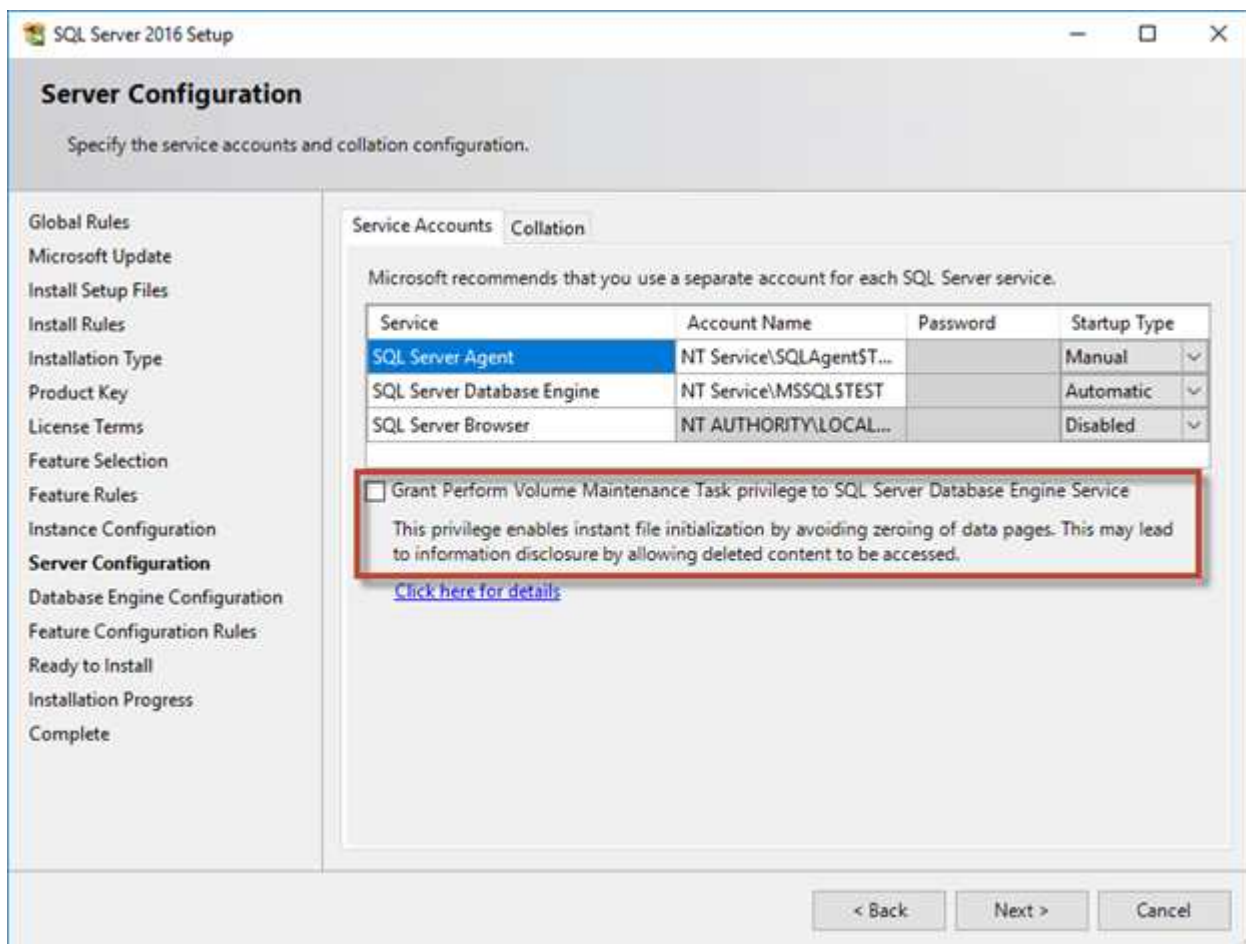
要检查权限是否已启用、您可以使用以下示例中的代码。此代码设置了两个跟踪标志、用于强制SQL Server将追加信息写入错误日志、创建小型数据库以及读取日志内容。

```
DBCC TRACEON(3004,3605,-1)
GO
CREATE DATABASE DelMe
GO
EXECUTE sp_readerrorlog
GO
DROP DATABASE DelMe
GO
DBCC TRACEOFF(3004,3605,-1)
GO
```

如果未启用即时文件初始化、则SQL Server错误日志会显示、除了将ldf日志文件置零之外、SQL Server还会将MDF数据文件置零、如以下示例所示。启用即时文件初始化后、它仅显示日志文件置零。

	LogDate	ProcessInfo	Text
365	2017-02-09 08:10:07.660	spid53	Ckpt dbid 3 flush delta counts.
366	2017-02-09 08:10:07.660	spid53	Ckpt dbid 3 logging active xact info.
367	2017-02-09 08:10:07.750	spid53	Ckpt dbid 3 phase 1 ended (8)
368	2017-02-09 08:10:07.750	spid53	About to log Checkpoint end.
369	2017-02-09 08:10:07.880	spid53	Ckpt dbid 3 complete
370	2017-02-09 08:10:08.130	spid53	Starting up database 'DelMe'.
371	2017-02-09 08:10:08.150	spid53	FixupLog Tail(progress) zeroing C:\Program Files\Micros
372	2017-02-09 08:10:08.160	spid53	Zeroing C:\Program Files\Microsoft SQL Server\MSSQ
373	2017-02-09 08:10:08.170	spid53	Zeroing completed on C:\Program Files\Microsoft SQL
374	2017-02-09 08:10:08.710	spid53	Ckpt dbid 6 started
375	2017-02-09 08:10:08.710	spid53	About to log Checkpoint begin.

执行卷维护任务在SQL Server 2016中得到了简化、稍后在安装过程中作为一个选项提供。此图显示了为SQL Server数据库引擎服务授予执行卷维护任务权限的选项。



控制数据库文件大小的另一个重要数据库选项是自动缩放。启用此选项后、SQL Server会定期缩减数据库文件、减小其大小并向操作系统释放空间。此操作会占用大量资源、并且很少有用、因为新数据进入系统一段时间后、数据库文件会再次增长。绝不能在数据库上启用自动缩放。

Microsoft SQL Server日志目录

日志目录在SQL Server中指定、用于在主机级别存储事务日志备份数据。如果使

用SnapCenter备份日志文件、则SnapCenter使用的每个SQL Server主机都必须配置一个主机日志目录、才能执行日志备份。SnapCenter 具有一个数据库存储库、因此与备份、还原或克隆操作相关的元数据存储在一个中央数据库存储库中。

主机日志目录的大小计算如下： 主机日志目录大小=((最大数据库LDF大小x每日日志更改率%) x (快照保留)÷(1 - LUN开销空间%)) 主机日志目录规模估算公式假定LUN开销空间为10%

将日志目录放在专用卷或LUN上。主机日志目录中的数据量取决于备份的大小以及备份的保留天数。SnapCenter只允许每个SQL Server主机有一个主机日志目录。您可以在SnapCenter → Host → Configure Plug-in中配置主机日志目录。



- NetApp建议*对主机日志目录执行以下操作：
- 确保主机日志目录不会被任何可能损坏备份快照数据的其他类型的数据共享。
- 请勿将用户数据库或系统数据库放置在托管挂载点的LUN上。
- 在SnapCenter将事务日志复制到的专用FlexVol卷上创建主机日志目录。
- 使用SnapCenter向导将数据库迁移到NetApp存储、以便数据库存储在有效位置、从而成功执行SnapCenter备份和还原操作。请注意、迁移过程会造成系统中断、并且在迁移过程中、发生原因会使数据库脱机。
- 对于SQL Server的故障转移集群实例(FCIS)、必须满足以下条件：
 - 如果您使用的是故障转移集群实例、则主机日志目录LUN必须是与要备份的SQL Server实例位于同一集群组中的集群磁盘资源SnapCenter。
 - 如果您使用的是故障转移集群实例、则必须将用户数据库置于共享LUN上、这些共享LUN是分配给与SQL Server实例关联的集群组的物理磁盘集群资源。

Microsoft SQL Server tempdb文件

Tempdb数据库的利用率可能会很高。除了在ONTAP上以最佳方式放置用户数据库文件之外、还可以更改tempdb数据文件以减少分配争用

如果SQL Server必须向特殊系统页面写入数据以分配新对象、则全局分配映射(GAM)、共享全局分配映射(SGAM)或页面可用空间(PFS)页面上可能会发生页面争用。页锁保护(锁定)内存中的这些页面。在繁忙的SQL Server实例上、可能需要很长时间才能在tempdb中的系统页面上获得锁定。这会导致查询运行时间变慢、并称为页锁争用。请参见以下创建tempdb数据文件的最佳实践：

- 对于<或=最多8个核心： tempdb数据文件=核心数
- 对于超过8个核心： 8个tempdb数据文件

以下示例脚本通过创建八个tempdb文件并将tempdb移动到挂载点来修改tempdb C:\MSSQL\tempdb 适用于SQL Server 2012及更高版本。

```
use master

go

-- Change logical tempdb file name first since SQL Server shipped with
```


logical file name called tempdev

```
alter database tempdb modify file (name = 'tempdev', newname =  
'tempdev01');
```

-- Change location of tempdev01 and log file

```
alter database tempdb modify file (name = 'tempdev01', filename =  
'C:\MSSQL\tempdb\tempdev01.mdf');
```

```
alter database tempdb modify file (name = 'templog', filename =  
'C:\MSSQL\tempdb\templog.ldf');
```

GO

-- Assign proper size for tempdev01

```
ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev01', SIZE = 10GB );
```

```
ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'templog', SIZE = 10GB );
```

GO

-- Add more tempdb files

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev02', FILENAME =  
N'C:\MSSQL\tempdb\tempdev02.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev03', FILENAME =  
N'C:\MSSQL\tempdb\tempdev03.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev04', FILENAME =  
N'C:\MSSQL\tempdb\tempdev04.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev05', FILENAME =  
N'C:\MSSQL\tempdb\tempdev05.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev06', FILENAME =  
N'C:\MSSQL\tempdb\tempdev06.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev07', FILENAME =  
N'C:\MSSQL\tempdb\tempdev07.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```
ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev08', FILENAME =  
N'C:\MSSQL\tempdb\tempdev08.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

从SQL Server 2016开始、安装期间会自动检测操作系统可见的CPU核心数、SQL Server会根据该数量计算和配置获得最佳性能所需的tempdb文件数。

Microsoft SQL Server和存储效率

ONTAP存储效率已进行优化、可存储和管理SQL Server数据、所用存储空间量最低、对系统整体性能几乎没有影响、甚至没有影响。

存储效率是RAID、配置(整体布局和利用率)、镜像和其他数据保护技术的组合。包括快照、精简配置和克隆在内的NetApp技术可优化基础架构中的现有存储、并推迟或避免未来的存储支出。将这些技术结合使用的次数越多、节省的空间就越大。

数据压缩、数据缩减和重复数据删除等空间效率功能旨在增加给定物理存储量所需的逻辑数据量。这样可以降低成本和管理开销。

从较高层面来看、数据压缩是一个数学过程、通过该过程、可以检测数据模式并对其进行编码、从而减少空间需求。相反、重复数据删除会检测实际重复的数据块并删除无关的副本。数据缩减允许多个逻辑数据块共享介质上的同一物理块。



有关存储效率与预留百分比之间交互的说明、请参见以下有关精简配置的章节。

压缩

在全闪存存储系统推出之前、基于阵列的数据压缩的价值有限、因为大多数I/O密集型工作负载都需要大量磁盘轴才能提供可接受的性能。由于驱动器数量众多、存储系统所含容量总是远远超出所需容量。随着固态存储的兴起、这种情况发生了变化。不再需要纯粹为了获得良好的性能而大量过度配置驱动器。存储系统中的驱动器空间可以与实际容量需求相匹配。

与旋转驱动器相比、固态驱动器(SSD)的IOPS功能提高几乎始终可以节省成本、但数据压缩可以通过增加固态介质的有效容量来进一步节省成本。

数据压缩方法有多种。许多数据库都具有自己的数据压缩功能、但在客户环境中很少出现这种情况。原因通常是对压缩数据*进行更改*会对性能造成影响、此外、对于某些应用程序、数据库级数据压缩的许可成本较高。最后、还会对数据库操作产生整体性能影响。为执行数据压缩和解压缩的CPU支付较高的每CPU许可证成本毫无意义、而不是实际的数据库工作。更好的选择是将压缩工作负载分流到存储系统。

自适应数据压缩

自适应数据压缩已针对企业级工作负载进行了全面测试、未观察到对性能的影响、即使在延迟以微秒为单位的全闪存环境中也是如此。一些客户甚至报告说、使用数据压缩后性能会提高、因为数据会在缓存中保持压缩状态、从而有效地增加了控制器中的可用缓存量。

ONTAP以4 KB为单位管理物理块。自适应数据压缩使用默认的压缩块大小8 KB、这意味着数据以8 KB单位进行压缩。这与关系数据库最常使用的8 KB块大小匹配。随着将更多数据作为一个单元进行压缩、数据压缩算法的效率也会提高。32 KB压缩块大小比8 KB压缩块单元更节省空间。这确实意味着、使用默认8 KB块大小的自适应数据压缩确实会使效率略低、但使用更小的数据压缩块大小也会有显著优势。数据库工作负载包含大量覆盖活动。要覆盖经过压缩的32 KB数据块中的8 KB、需要回读整个32 KB逻辑数据、对其进行解压缩、更新所需的8 KB区域、重新压缩、然后将整个32 KB写入驱动器。这对存储系统来说是一项非常昂贵的操作、因此、某些基于

较大压缩块大小的竞争存储阵列也会对数据库工作负载的性能造成严重影响。



自适应数据压缩使用的块大小最多可以增加至32 KB。这可能会提高存储效率、对于事务日志和备份文件等不活动的文件、如果阵列上存储了大量此类数据、则应考虑使用此方法。在某些情况下、使用16 KB或32 KB块大小的活动数据库也可以通过增加要匹配的自适应数据压缩的块大小来受益。请咨询NetApp或合作伙伴代表、了解这是否适合您的工作负载。



在流式备份目标上、不应同时使用大于8 KB的数据压缩块大小和重复数据删除。原因是、对备份的数据所做的微小更改会影响32 KB数据压缩窗口。如果窗口发生变化、则生成的压缩数据会在整个文件中有所不同。重复数据删除在数据压缩后进行、这意味着重复数据删除引擎对每个压缩备份的看法不同。如果需要对流式备份进行重复数据删除、则只应使用8 KB块自适应数据压缩。最好使用自适应数据压缩、因为它的块大小较小、不会影响重复数据删除的效率。出于类似的原因、主机端压缩也会影响重复数据删除效率。

数据压缩对齐

数据库环境中的自适应数据压缩需要在一定程度上考虑数据压缩块对齐问题。对于随机覆盖非常特定的块的数据来说、这样做只是一个问题。这种方法在概念上类似于整体文件系统对齐、即文件系统的起点必须与4 k设备边界对齐、文件系统的块大小必须是4 k的倍数。

例如、只有当8 KB写入文件与文件系统本身内的8 KB边界对齐时、才会对其进行压缩。这一点意味着它必须位于文件的前8 KB、文件的后8 KB、依此类推。要确保正确对齐、最简单的方法是使用正确的LUN类型、创建的任何分区都应与设备起始位置偏移8K的倍数、并使用数据库块大小的倍数作为文件系统块大小。

备份或事务日志等数据是跨多个块按顺序写入的操作、所有这些块都会进行压缩。因此、无需考虑对齐。唯一关注的I/O模式是随机覆盖文件。

数据缩减

数据缩减是一项可提高数据压缩效率的技术。如前文所述、自适应数据压缩本身最多可节省2: 1的空间、因为它仅限于在4 KB WAFL块中存储8 KB I/O。块大小越大、压缩方法的效率越高。但是、它们不适用于受到小块覆盖的数据。解压缩32 KB数据单元、更新8 KB部分、重新压缩以及回写驱动器会产生开销。

数据缩减的工作原理是、允许将多个逻辑块存储在物理块中。例如、具有高度可压缩数据(例如文本或部分全满块)的数据库可以从8 KB压缩到1 KB。如果不进行数据缩减、这1 KB的数据仍会占用整个4 KB块。实时数据缩减允许将1 KB的压缩数据与其他压缩数据一起存储在仅1 KB的物理空间中。它不是一种压缩技术; 它只是一种在驱动器上分配空间的更高效的方式、因此不会产生任何可检测的性能影响。

节省的资金数额各不相同。已压缩或加密的数据通常无法进一步压缩、因此、数据集无法从数据缩减中受益。相比之下、新初始化的数据文件包含的块元数据和零数据略多、数据压缩率高达80: 1。

对温度敏感的存储效率

温度敏感型存储效率(TSSE)在ONTAP 9.8及更高版本中提供、它依靠块访问热图来识别不常访问的块并以更高的效率对其进行压缩。

重复数据删除

重复数据删除是指从数据集中删除重复的块大小。例如、如果10个不同文件中存在相同的4 KB块、则重复数据删除会将所有10个文件中的4 KB块重定向到相同的4 KB物理块。结果是、这些数据的效率将提高10: 1。

VMware子系统启动LUN等数据的重复数据删除效果通常非常好、因为它们包含同一操作系统文件的多个副本。

我们观察到的效率为100:1甚至更高。

某些数据不包含重复数据。例如、Oracle块包含数据库全局唯一的标头和几乎唯一的尾部。因此、对Oracle数据库进行重复数据删除很少能节省超过1%的空间。对MS SQL数据库执行重复数据删除略有改进、但块级别的唯一元数据仍是一个限制。

在少数情况下、使用16 KB和大型块的数据库可节省多达15%的空间。每个块的初始4 KB包含全局唯一标头、而最后4 KB块包含接近唯一的尾部。内部块是重复数据删除的候选数据、但实际上、这几乎完全是由于对置零数据进行重复数据删除。

许多争用资源的阵列都声称可以根据数据库被复制多次的假设对数据库进行重复数据删除。在这方面、也可以使用NetApp重复数据删除、但ONTAP提供了一个更好的选择: NetApp FlexClone技术。最终结果是相同的; 系统会为一个数据库创建多个副本、这些副本共享大多数底层物理块。与花时间复制数据库文件并对其进行重复数据删除相比、使用FlexClone的效率要高得多。实际上、它是无重复数据删除、而不是重复数据删除、因为从一开始就不会创建重复数据。

效率和精简配置

效率功能是精简配置的一种形式。例如、占用100 GB卷的100 GB LUN可能会压缩到50 GB。由于卷仍为100 GB、因此尚未实现实际节省。必须先减小卷大小、以便节省的空间可用于系统上的其他位置。如果稍后更改100 GB LUN会导致数据的可压缩性降低、则LUN大小会增大、卷可能会填满。

强烈建议使用精简配置、因为它可以简化管理、同时显著提高可用容量并节省相关成本。原因很简单—数据库环境通常包含大量空空间、大量卷和LUN以及可压缩数据。厚配置会为卷和LUN预留存储空间、以防它们最终达到100%全满并包含100%不可压缩数据。这种情况不大可能发生。通过精简配置、可以回收这些空间并将其用于其他位置、并可以基于存储系统本身进行容量管理、而不是基于许多较小的卷和LUN。

有些客户更喜欢对特定工作负载使用厚配置、或者通常根据既定的运营和采购实践使用厚配置。

***注意:** *如果卷配置厚配置、则必须小心操作、以便完全禁用该卷的所有效率功能、包括使用解压缩和删除重复数据删除 `sis undo` 命令: 此卷不应显示在中 `volume efficiency show` 输出。如果配置了效率功能、则仍会为卷部分配置效率功能。因此、覆盖保证的工作方式有所不同、这会增加配置忽略发生原因卷以意外用尽空间的可能性、从而导致数据库I/O错误。

效率最佳实践

NetApp建议执行以下操作:

AFF默认值

在纯闪存AFF系统上运行的ONTAP上创建的卷经过精简配置、并启用了所有实时效率功能。尽管数据库通常不会从重复数据删除中受益、并且可能包含不可压缩的数据、但默认设置适用于几乎所有工作负载。ONTAP旨在高效处理所有类型的数据和I/O模式、无论它们是否可节省空间。只有在完全了解原因且有优势的情况下、才应更改默认值。

一般建议

- 如果卷和(或) LUN未进行精简配置、则必须禁用所有效率设置、因为使用这些功能不会节省空间、而将厚配置与已启用空间效率相结合会发生原因发生意外行为、包括空间不足错误。
- 如果数据不会被覆盖(例如使用备份或数据库事务日志)、则可以通过在较低的冷却期启用TSSE来提高效率。
- 某些文件可能包含大量不可压缩数据、例如、在应用程序级别已启用数据压缩时、文件已加密。如果出现上述任一情况、请考虑禁用数据压缩、以便在包含可压缩数据的其他卷上执行更高效的操作。

- 不要在数据库备份中同时使用32 KB数据压缩和重复数据删除。请参见一节 [\[自适应数据压缩\]](#) 了解详细信息。

数据库压缩

SQL Server本身还具有压缩和高效管理数据的功能。SQL Server目前支持两种类型的数据压缩：行压缩和页面压缩。

行压缩会更改数据存储格式。例如、它会将整数和小数更改为可变长度格式、而不是其本机固定长度格式。此外、它还会通过消除空格将固定长度字符串更改为可变长度格式。页面压缩可实现行压缩和另外两种压缩策略(前缀压缩和词典压缩)。有关页面压缩的详细信息、请参见 "[页面压缩实施](#)"。

SQL Server 2008及更高版本的企业版、开发人员版和评估版目前支持数据压缩。虽然数据压缩可以由数据库本身执行、但在SQL Server环境中很少会出现这种情况。

下面是管理SQL Server数据文件空间的建议

- 在SQL Server环境中使用精简配置可提高空间利用率、并在使用空间保证功能时降低整体存储需求。
- 对最常见的部署配置使用自动增长、因为存储管理员只需要监控聚合中的空间使用量。
- 建议不要在包含SQL Server数据文件的任何卷上启用重复数据删除、除非该卷已知包含同一数据的多个副本、例如、将数据库从备份还原到单个卷。

空间回收

可以定期启动空间回收、以恢复LUN中未使用的空间。对于SnapCenter、您可以使用以下PowerShell命令启动空间回收。

```
Invoke-SdHostVolumeSpaceReclaim -Path drive_path
```

如果需要运行空间回收、则应在活动较少的时段运行此过程、因为它最初会占用主机上的周期。

使用NetApp管理软件保护Microsoft SQL Server数据

根据业务需求规划数据库备份。通过将ONTAP的NetApp Snapshot技术与Microsoft SQL Server API相结合、无论用户数据库大小如何、您都可以快速创建应用程序一致的备份。为了满足更高级或横向扩展的数据管理要求、NetApp提供了SnapCenter。

SnapCenter

SnapCenter是一款适用于企业级应用程序的NetApp数据保护软件。使用适用于SQL Server的SnapCenter插件以及适用于Microsoft Windows的SnapCenter插件管理的操作系统、可以快速轻松地保护SQL Server数据库。

SQL Server实例可以是独立设置、故障转移集群实例、也可以始终位于可用性组上。这样、可以通过单一管理平台从主副本或二级副本保护、克隆和还原数据库。SnapCenter可以管理内部、云端和混合配置中的SQL Server数据库。此外、还可以在原始主机或备用主机上几分钟内创建数据库副本、以供开发或报告之用。



* NetApp建议*使用SnapCenter创建Snapshot副本。下文所述的T-SQL方法也有效、但SnapCenter可实现备份、还原和克隆过程的完全自动化。同时、它还会执行发现以确保创建的快照正确无误。无需预配置。... SQL Server还要求操作系统和存储之间进行协调、以确保创建快照时Snapshot中存在正确的数据。在大多数情况下、执行此操作的唯一安全方法是使用SnapCenter或T-SQL。如果未进行这种额外协调、则创建的快照可能无法可靠地恢复。

有关适用于SnapCenter的SQL Server插件的详细信息、请参见 ["TR-4714: 《使用NetApp SnapCenter的SQL Server最佳实践指南》"](#)。

使用T-SQL快照保护数据库

在SQL Server 2022中、Microsoft引入了T-SQL快照、可通过它编写脚本并自动执行备份操作。您可以为快照准备数据库、而不是执行全尺寸副本。数据库备份就绪后、您可以利用ONTAP REST API创建快照。

以下是备份工作流示例：

1. 使用alter命令冻结数据库。这样可以使数据库在底层存储上为一致的快照做好准备。冻结后、您可以解冻数据库并使用backup命令记录快照。
2. 使用新的backup group和backup server命令同时对存储卷上的多个数据库执行快照。
3. 执行完整备份或copy_only完整备份。这些备份也会记录在msdb中。
4. 在快照完整备份之后、使用通过正常流式传输方法创建的日志备份执行时间点恢复。如果需要、还支持流式差异备份。

要了解更多信息、请参见 ["了解T-SQL快照的Microsoft文档"](#)。

使用ONTAP进行Microsoft SQL Server灾难恢复

企业数据库和应用程序基础架构通常需要进行复制、以防止发生自然灾害或意外业务中断、同时最大限度地减少停机时间。

SQL Server无中断可用性组复制功能是一个绝佳的选择、NetApp提供了将数据保护与无中断集成的选项。但是、在某些情况下、您可能需要考虑使用ONTAP复制技术。ONTAP复制选项(包括MetroCluster和SnapMirror)可以更好地扩展、对性能的影响微乎其微、保护非SQL数据、并且通常可提供完整基础架构复制和灾难恢复解决方案。

SnapMirror异步

SnapMirror技术提供了一个快速灵活的异步企业解决方案、用于通过LAN和广域网复制数据。在创建初始镜像后、SnapMirror技术仅会将更改后的数据块传输到目标、从而显著降低网络带宽要求。

下面是有关适用于SQL Server的SnapMirror的建议：

- 如果使用CIFS、则目标SVM必须是源SVM所属的同一Active Directory域的成员、以便在灾难恢复期间不会破坏NAS文件中存储的访问控制列表(ACL)。
- 使用与源卷名称相同的目标卷名称并不是必需的、但这样可以简化将目标卷挂载到目标的过程、使其易于管理。如果使用CIFS、则必须使目标NAS命名空间与源命名空间的路径和目录结构完全相同。
- 为保持一致、请勿从控制器计划SnapMirror更新。而是应启用SnapCenter中的SnapMirror更新、以便在完整备份或日志备份完成后更新SnapMirror。

- 将包含SQL Server数据的卷分布在集群中的不同节点上、以允许所有集群节点共享SnapMirror复制活动。此分布可优化节点资源的使用。

有关SnapMirror的详细信息、请参见 ["TR-4015: 《适用于ONTAP 9的SnapMirror配置和最佳实践指南》"](#)。

保护ONTAP上的Microsoft SQL Server安全

保护SQL Server数据库环境的安全不仅仅是管理数据库本身、还需要进行多层面的努力。ONTAP提供了多种独特的功能、旨在保护数据库基础架构的存储方面。

Snapshot 副本

存储快照是目标数据的时间点副本。ONTAP的实施支持设置各种策略、并为每个卷存储多达1024个快照。ONTAP中的快照可节省空间。只有在原始数据集发生更改时、才会占用空间。它们也是只读的。快照可以删除、但不能更改。

在某些情况下、可以直接在ONTAP上计划快照。在其他情况下、可能需要SnapCenter等软件来编排应用程序或操作系统操作、然后再创建快照。无论哪种方法最适合您的工作负载、主动快照策略都可以通过频繁、轻松访问从启动LUN到任务关键型数据库的所有备份来确保数据安全。

注意：ONTAP灵活卷或更简单地说、卷不是LUN的同义词。卷是文件或LUN等数据的管理容器。例如、数据库可能位于一个8 LUN条带集上、所有LUN都包含在一个卷中。

有关快照的详细信息、请单击 ["此处。"](#)

防篡改快照

从ONTAP 9.12.1开始、快照不仅是只读的、还可以防止意外或故意删除。此功能称为防篡改快照。可以通过Snapshot策略设置和实施保留期限。生成的快照在达到到期日期之前无法删除。不会覆盖管理或支持中心。

这样可以确保即使侵入者、恶意内部人员甚至勒索软件攻击导致访问ONTAP系统本身、也不会影响备份。如果与频繁的快照计划结合使用、则可以提供极其强大的数据保护、并且RPO非常低。

有关防篡改快照的详细信息、请单击 ["此处。"](#)

SnapMirror 复制

也可以将快照复制到远程系统。这包括防篡改快照、其中会在远程系统上应用和实施保留期限。结果是、数据保护优势与本地快照相同、但数据位于另一个存储阵列上。这样可以确保销毁原始阵列不会影响备份。

第二个系统还为管理安全性打开了新的选项。例如、某些NetApp客户会隔离主存储系统和二级存储系统的身份验证凭据。没有一个管理用户可以同时访问这两个系统、这意味着恶意管理员无法删除所有数据副本。

有关SnapMirror的详细信息、请单击 ["此处。"](#)

Storage Virtual Machine

新配置的ONTAP存储系统与新配置的VMware ESX服务器类似、因为在创建虚拟机之前、这两个系统都不能支持任何用户。通过ONTAP、您可以创建一个Storage Virtual Machine (SVM)、此SVM将成为存储管理的最基本单元。每个SVM都有自己的存储资源、协议配置、IP地址和FCP WWN。这是ONTAP多租户的基础。

例如、您可以为关键生产工作负载配置一个SVM、为开发活动配置另一个SVM。然后、您可以限制某些管理员对生产SVM的访问、同时授予开发人员对开发SVM中存储资源的更广泛的控制权限。您可能还需要为财务和人力资源团队提供第三个SVM、以存储非常关键的只读数据。

有关SVM的详细信息、请单击 ["此处。"](#)

管理RBAC

ONTAP为管理登录提供了功能强大的基于角色的访问控制(Role-Based Access Control、RBAC)。某些管理员可能需要完全集群访问权限、而其他管理员则可能只需要访问某些SVM。高级帮助台人员可能需要增加卷大小的能力。这样、您就可以为管理用户授予执行其工作职责所需的访问权限、而无需执行任何其他操作。此外、您还可以使用不同供应商提供的PKI来保护这些登录、限制对ssh密钥的访问、并强制执行失败登录尝试锁定。

有关管理访问控制的详细信息、请单击 ["此处。"](#)

多因素身份验证

ONTAP和某些其他NetApp产品现在支持使用多种方法进行多因素身份验证(MFA)。其结果是、用户名/密码被泄露本身并不是没有第二个因素(例如FOB或智能手机应用程序)的数据的安全线程。

有关详细信息、请单击 ["此处。"](#)

API RBAC

自动化需要API调用、但并非所有工具都需要完全管理访问权限。为了帮助保护自动化系统的安全、还在API级别提供了RBAC。您可以将自动化用户帐户限制为所需的API调用。例如、监控软件不需要更改访问权限、只需要读取访问权限即可。配置存储的工作流不需要删除存储。

要了解更多信息、请启动https://docs.netapp.com/us-en/ontap-automation/rest/rbac_overview.html[here.]

多管理员验证(MAV)

通过要求两个不同的管理员(每个管理员都有自己的凭据)批准某些活动、可以进一步实现多"因素"身份验证。其中包括更改登录权限、运行诊断命令和删除数据。

有关多管理员验证(MAV)的详细信息、请单击 ["此处"](#)

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。