



# Red Hat Directory Server 11

## 性能调优指南

调整目录服务器的性能



## 调整目录服务器的性能

Marc Muehlfeld

Red Hat Customer Content Services

Petr Bokoč

Red Hat Customer Content Services

Tomáš Čapek

Red Hat Customer Content Services

Ella Deon Ballard

Red Hat Customer Content Services

## 法律通告

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

本指南为提高服务器和数据库性能提供了提示。

# 目录

使开源更具 INCSIVE .....	3
<b>第 1 章 DIRECTORY 服务器性能调优简介 .....</b>	<b>4</b>
1.1. 为 DIRECTORY 服务器性能设置目标 .....	4
<b>第 2 章 跟踪服务器和数据库性能 .....</b>	<b>6</b>
2.1. 监控服务器活动 .....	6
2.2. 监控数据库活动 .....	9
2.3. 监控数据库链接活动 .....	13
2.4. 监控本地磁盘 GRACEFUL SHUTDOWN .....	14
<b>第 3 章 调整锁定数 .....</b>	<b>16</b>
3.1. 手动监控锁定数 .....	16
3.2. 通过监控自由数据库锁定来避免数据崩溃 .....	16
3.3. 使用命令行设置锁定数 .....	16
3.4. 使用 WEB 控制台设置锁定数 .....	17
<b>第 4 章 提高搜索性能（及平衡读取性能） .....</b>	<b>18</b>
4.1. 使用索引 .....	18
4.2. 调优 DIRECTORY 服务器资源设置 .....	19
4.3. 设置索引扫描限制 .....	21
4.4. 精细约束 ID 列表大小 .....	22
4.5. 为搜索调整数据库缓存 .....	23
4.6. 管理特殊条目 .....	23
<b>第 5 章 调整事务日志记录 .....</b>	<b>24</b>
5.1. 将数据库目录移到 9 月磁盘或分区中 .....	24
5.2. 更改数据库检查点 INTERVAL .....	25
5.3. 禁用 DURABLE TRANSACTIONS .....	26
5.4. 指定事务批处理 .....	26
<b>第 6 章 管理数据库缓存设置 .....</b>	<b>27</b>
6.1. DATABASE 和 ENTRY CACHE AUTO-SIZING 功能 .....	27
6.2. 确定所需的缓存大小 .....	29
6.3. 手动设置 ENTRY CACHE SIZE .....	31
6.4. 设置 DN 缓存的大小 .....	33
6.5. 设置数据库缓存大小 .....	34
<b>第 7 章 设置目录服务器线程的数量 .....</b>	<b>38</b>
7.1. 自动线程调优 .....	38
7.2. 手动设置线程数 .....	40
<b>第 8 章 调优复制性能 .....</b>	<b>42</b>
8.1. 提高多层次复制效率 .....	42
<b>第 9 章 调优数据库链路性能 .....</b>	<b>44</b>
9.1. 管理到远程服务器的连接 .....	44
9.2. 在正常处理过程中检测错误 .....	47
<b>第 10 章 提高导入性能 .....</b>	<b>49</b>
10.1. 为大型数据库导入和利用大量属性导入来调优目录服务器 .....	49
10.2. 调优目录服务器以导入大量条目 .....	49
<b>附录 A. 修订历史记录 .....</b>	<b>50</b>



---

## 使开源更具 INCSIVE

红帽致力于替换我们的代码、文档和 Web 属性中有问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [“CTO Chris Wright”](#) 的信息。

## 第 1 章 DIRECTORY 服务器性能调优简介

本文提供了一些流程和选项，管理员可用于优化其红帽目录服务器部署的性能。性能调优目录服务器实例对每台服务器而言是唯一的，因为对机器环境中的每台服务器、目录大小和数据类型、负载和网络使用情况的差异，即使用户和客户端执行的操作类型也是如此。

本指南的目的是强调红帽目录服务器为跟踪和评估服务器和数据库性能提供的功能。还提供了一些有助于调优服务器性能的步骤。但是，请参阅红帽目录服务器 [管理指南](#)，[查看红帽目录服务器部署指南](#)，以及有关基于命令行和 UI 的管理命令，请参阅 [红帽目录服务器管理指南](#)。

### 1.1. 为 DIRECTORY 服务器性能设置目标

性能调优只是识别服务器正常操作环境中潜在的（或真实）瓶颈的方法，然后采取措施缓解这些瓶颈。

性能调优的一般计划是：

1. 评估环境。查看目录服务器的一切信息：其使用情况、负载、网络连接和可靠性、最常见的操作、其物理机器及其所在的物理机器，以及竞争其资源的所有服务。
2. 测量当前目录服务器性能并建立基线。
3. 识别可改进的服务器区域。
4. 对 Directory 服务器设置进行任何更改，并可能对主机机器进行修改。
5. 再次测量 Directory 服务器性能，以查看影响性能的变化。

目录服务器在三个区域提供某种监控：

- 服务器进程（计数和日志）
- 数据库（计数器）
- 任何数据库链接（计数）

在 Directory Server 中，大多数性能测量将如何检索并提供给客户端。考虑到这一点，它们是可以针对最佳 Directory 服务器性能调整的服务器区域（以及本文中涵盖的领域）：

- 搜索操作
- 索引性能（影响搜索和写操作）
- 数据库事务
- 数据库和条目缓存设置
- 数据库链接

可以对主机机器的设置或硬件进行其他更改，也可以影响 Directory 服务器性能：

- 可用内存（基于目录大小）
- 在同一计算机上运行的其他服务器（资源竞争）
- 在其他机器上的其他目录服务器实例中分发用户数据库
- 由于网络性能而平衡服务器负载



与可以对实例所做的更改相比，这些更改更多地规划有效目录服务器部署。《部署指南》可以提供更多有关如何规划最佳企业部署的详细信息。

## 第 2 章 跟踪服务器和数据库性能

红帽目录服务器有两种记录和跟踪性能数据的方法：性能计数器和日志。计数器用于确定目录服务器的性能如何，特别是数据库性能；日志用于诊断服务器和 LDAP 操作和配置的任何问题区域。

性能计数器专注于服务器、所有配置的数据库和数据库链接（链数据库）的操作和信息。

有三种日志类型：访问（用于客户端连接）、错误（用于错误、警告和事件详情）和 audit（更改目录服务器配置）。默认运行的访问和错误日志（以及服务器运行需要错误日志）。由于开销，必须手动启用审计日志记录。



### 注意

访问日志已被缓冲。即使加载的服务器也允许完全访问记录，但在服务器中发生事件以及事件写入日志时之间会有一个时间滞后。

### 2.1. 监控服务器活动

可通过 Web 控制台或命令行监控目录服务器的当前活动。也可以监控所有数据库缓存的活动。



### 注意

由服务器监控的目录服务器数据库属性的一些计数器使用 64 位整数，即使在 32 位系统中，即使是 32 位系统（启动的连接、操作完成、操作完成、发送条目和字节数发送）。在高容量系统上，这可防止滚动的计数器的速度过快，并造成监控数据。

#### 2.1.1. 使用命令行监控目录服务器

使用命令行监控服务器：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com monitor server
```

下表描述了命令返回的属性：

表 2.1. 服务器监控属性

属性	描述
version	标识目录的当前版本号。
threads	用于处理请求的当前活动线程数量。额外的线程可以由内部服务器任务创建，如复制或串联。

属性	描述
连接	<p>为每个打开的连接提供以下摘要信息（仅在作为 Directory Manager 的绑定到目录时才可用）：</p> <div style="border: 1px solid black; padding: 5px;"> <p><i>fd</i> - 此连接的文件描述符。</p> <p><i>opentime</i> - 打开此连接的时间。</p> <p><i>opsinitiated</i> - 此连接启动的操作数量。</p> <p><i>opscompleted</i> - 已完成的操作数量。</p> <p><i>bindDN</i> - 此连接用于连接的目录的可分辨名称。</p> <p><i>rw</i> - 如果连接被阻止进行读取或写入，显示的字段。</p> </div> <p>默认情况下，此信息可供 Directory Manager 使用。但是，可以编辑与此信息关联的 ACI，以允许其他人访问信息。</p>
currentconnections	按目录确定服务当前在服务中的连接数。
totalconnections	识别从目录启动以来所处理的连接数量。
currentconnectionsatmaxthreads	显示所有当前处于 <b>最大线程</b> 状态的连接。
maxthreadsperrconnhits	显示连接达到最大 <b>线程的次数</b> 。
dtablesize	显示可用于目录的文件描述符数量。每个连接都需要一个文件描述符：每个打开的索引对应一个，一个用于日志文件管理，另一个用于 <b>ns-slapd</b> 本身。本质上，这个值显示目录中可以服务多少个额外的并发连接。有关文件描述符的更多信息，请参阅操作系统文档。
readwaiters	标识等待从客户端读取数据的线程数量。
opsinitiated	标识服务器启动以来启动的操作数量。
opscompleted	标识服务器启动以来已完成的操作数量。
entriessent	标识在服务器启动时发送到客户端的条目数量。
bytessent	标识在服务器启动后发送到客户端的字节数。
currenttime	标识执行服务器快照的时间。时间以 UTC 格式显示 Greenwich Mean Time(GMT)。

属性	描述
starttime	标识服务器启动的时间。时间以 UTC 格式显示 Greenwich Mean Time(GMT)。
nbackends	标识服务器服务的后端数量（数据库）。

## 2.1.2. 使用 Web 控制台监控服务器

使用 Web 控制台监控服务器：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择实例。
3. 在 **Monitoring** 选项卡中，选择 **Server stats**。

The screenshot shows the 'Monitoring' section of the web console. On the left is a navigation menu with 'Server Statistics' selected. The main content area shows 'Server Statistics' for the instance 'slapd-sample\_instance'. Below this, there are tabs for 'Resource Charts', 'Server Stats', 'Connection Table', 'Disk Space', and 'SNMP Counters'. The 'Server Stats' tab is active, displaying a table of server metrics:

Server Instance	slapd-sample_instance		
Version	389-Directory/1.4.3.31 B2022.292.1953		
Server Started	2022/10/24, 11:16:11		
Server Uptime	21 days, 3 hours, 9 minutes, and 44 seconds		
Worker Threads	16	Threads Waiting To Read	2
Conns At Max Threads	0	Conns Exceeded Max Threads	0
Total Connections	15	Current Connections	3
Operations Started	63	Operations Completed	60
Entries Returned To Clients	32	Bytes Sent to Clients	460890

下表描述了此菜单中可见的字段：

表 2.2. 常规信息（服务器）

字段	描述
服务器实例	显示 Directory 服务器实例的名称。
版本	标识当前服务器版本。
服务器已启动	服务器启动的日期和时间。
server Uptime	实例运行的时间。

字段	描述
Worker Threads	用于处理请求的当前活动线程数量。额外的线程可以由内部服务器任务创建，如复制或串联。
线程等待读取	等待从客户端读取的线程总数。如果服务器开始从客户端接收请求，则线程可能不会立即读取，然后因某种原因停止该请求的传输。通常，等待读取的线程表示网络或客户端慢的问题。
conns At Max Threads	显示所有当前处于 <b>最大线程</b> 状态的连接。
conns Hit Max Threads	显示连接达到最大 <b>线程的次数</b> 。
连接总数	为这个目录服务器实例建立的连接总数。
当前连接	开放连接总数。每个连接都可以考虑多个操作，因此多个线程。
操作已启动	此连接启动的操作数量。
操作完成	服务器为此连接完成的操作数量。
返回客户端的条目	自服务器启动以来发送到客户端的条目数量。
客户端的字节数	自服务器启动以来发送到客户端的字节数。

## 2.2. 监控数据库活动



### 注意

由服务器监控的目录服务器数据库属性的一些计数器使用 64 位整数，即使在 32 位系统中（条目缓存点击、条目缓存尝试、当前缓存大小和最大缓存大小）。在高容量系统上，这可防止滚动的计数器的速度过快，并造成监控数据。

### 2.2.1. 使用命令行监控数据库活动

监控数据库的当前活动：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com monitor backend
```

下表描述了命令返回的属性：

表 2.3. 数据库监控属性

属性	描述
ReadOnly	指明数据库是否处于只读模式(1)还是读写模式(0)。

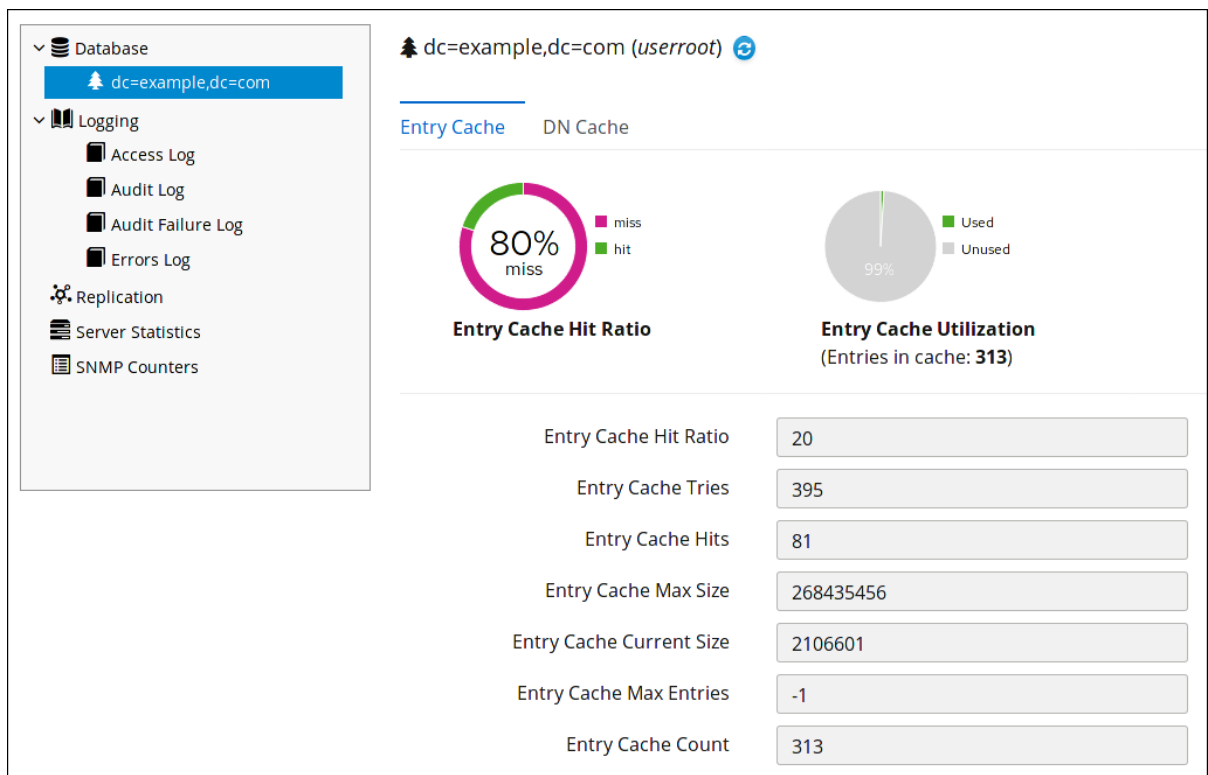
属性	描述
entrycachehits	成功条目缓存查找的总数。该值是服务器从条目缓存检索条目的次数，而不必从数据库重新加载它。
entrycachetries	从您启动实例以来，条目缓存查找总数。值是实例启动以来的总数，{DS} 尝试从条目缓存中检索条目。
entrycachehitratio	<p>条目缓存会尝试成功条目缓存查找的数量。此数字基于自上次启动实例以来的总查询和点击。条目缓存的点击率越接近为 100%，这越好。</p> <p>每当操作尝试查找条目查询时，服务器都需要访问数据库来获取条目。因此，随着这种比例下降到零，磁盘访问数量会增加，并且目录搜索性能会降低。要提高此比率，增大数据库条目缓存的大小。</p> <p>要提高这个比例，请通过增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-cachememsize</b> 属性的值来增加条目缓存的大小。</p>
currententrycachesize	<p>条目缓存中当前存在的目录条目的总大小（以字节为单位）。</p> <p>要增加缓存中可以存在的条目的大小，请增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-cachememsize</b> 属性的值。</p>
maxentrycachesize	<p>{DS} 可在条目缓存中维护的目录条目的最大大小，以字节为单位。</p> <p>要增加缓存中可以存在的条目的大小，请增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-cachememsize</b> 属性的值。</p>
currententrycachecount	存储在给定后端条目缓存中的当前条目数。
maxentrycachecount	<p>存储在数据库条目缓存中的最大条目数。</p> <p>要调整这个值，请增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 中的 <b>nsslapd-cachesize</b> 属性的值</p>
dncachehits	服务器通过从 DN 缓存中获取规范化区分名称(DN)来处理请求的次数，而不是对它进行规范化。
dncachetries	自启动实例以来，DN 缓存访问总数。
dncachehitratio	缓存的比例试图成功进行 DN 缓存命中。更接近这个值是 100%，越好。
currentdncachesize	<p>DN 缓存中当前存在的 DN 的总大小（以字节为单位）。</p> <p>要增加 DN 缓存中可以存在的条目的大小，请增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-dncachememsize</b> 属性的值。</p>

属性	描述
maxdncachesize	{DS} 可以在 DN 缓存中维护的最大大小，以字节为单位。 要增加缓存中可以存在的条目的大小，请增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-dncachememsize</b> 属性的值。
currentdncachecount	DN 缓存中当前存在的 DN 数量。
maxdncachecount	DN 缓存中允许的最大 DN 数量。

## 2.2.2. 使用 Web 控制台监控数据库活动

使用 Web 控制台监控数据库活动：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 *Red Hat Directory Server Administration Guide* 中的 [使用 Web 控制台登录到目录服务器](#) 一节。
2. 选择实例。
3. 在 **Monitoring** 选项卡上，选择要显示的数据库条目。
4. 选择 **Entry Cache** 以显示条目缓存的性能值：



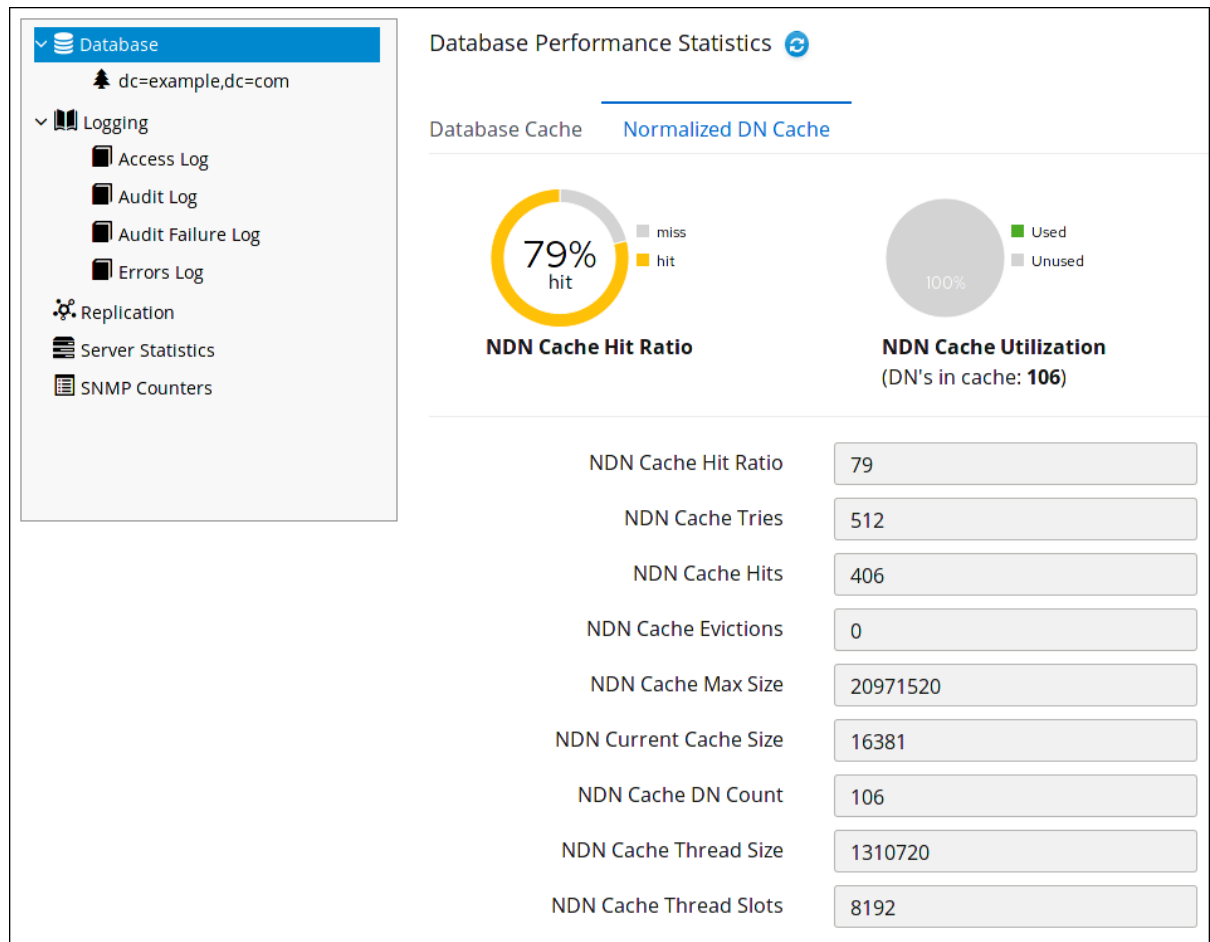
下表描述了此标签页中可见的字段：

表 2.4. Entry Cache 选项卡上的字段

字段名称	描述
entry Cache Hit Ratio	<p>表示条目缓存次数的比例是成功条目缓存查找。此数字基于自目录上次启动以来的总查询和点击。更接近这个值是100%，越好。每当操作试图查找条目在条目缓存中时，目录都必须执行磁盘访问权限来获取条目。因此，随着这种比例下降到零，磁盘访问数量会增加，并且目录搜索性能会下降。</p> <p>要提高这个比例，请通过增加 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-cachememsize</b> 属性的值来增加条目缓存的大小。</p>
entry Cache Tries	从目录上次启动以来条目缓存查找的总数。也就是说，自服务器启动以来请求的条目总数。
条目缓存 Hits	成功条目缓存查找的总数。也就是说，服务器可以通过从缓存中获取数据而不是访问磁盘来处理搜索请求的次数。
条目缓存最大大小	<p>条目缓存的大小（以字节为单位），由目录维护。</p> <p>这个值由数据库的 <b>cn=database_name,cn=ldbm database,cn=plugins,cn=config</b> 条目中的 <b>nsslapd-cachememsize</b> 属性管理。</p>
条目缓存当前大小	条目缓存中当前存在的目录条目数。
条目缓存最大条目	<p><i>已弃用。</i></p> <p>在条目缓存中维护的最大目录条目数。</p> <p>不要尝试通过设置最大允许条目数来管理缓存大小。这使得主机难以有效地分配 RAM。通过使用 <b>nsslapd-cachememsize</b> 属性，通过设置缓存的可用 RAM 数量来管理缓存大小。</p>
条目缓存数	条目缓存中当前存在的目录条目数。

#### 5. 为 DN 缓存的性能值选择 **DN Cache**。





### 2.3. 监控数据库链接活动

也可以显示数据库链接（链数据库）的活动，但只能使用命令行：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com monitor chaining
```

下表描述了命令返回的属性：

表 2.5. 数据库链路监控属性

属性名称	描述
nsAddCount	接收的添加操作数量。
nsDeleteCount	收到删除操作的数量。
nsModifyCount	收到修改操作的数量。
nsRenameCount	收到的重命名操作数量。
nsSearchBaseCount	收到的基础级搜索数量。
nsSearchOneLevelCount	收到的一级搜索数。
nsSearchSubtreeCount	接收的子树搜索数量。

属性名称	描述
nsAbandonCount	接收的带外操作数。
nsBindCount	收到的绑定请求数量。
nsUnbindCount	收到的未绑定数量。
nsCompareCount	收到的比较操作数量。
nsOperationConnectionCount	正常操作的打开连接数量。
nsBindConnectionCount	绑定操作的开放连接数。

## 2.4. 监控本地磁盘 GRACEFUL SHUTDOWN

当系统中的磁盘空间太小时，目录服务器进程会终止。因此，数据库或松散数据存在损坏风险。

要防止这个问题，您可以配置 Directory 服务器来监控可用磁盘空间。监控线程检查包含配置、事务日志和数据库目录的文件系统上的可用空间。

根据剩余的可用磁盘空间，Directory 服务器的行为有所不同：

- 如果可用磁盘空间达到定义的阈值，Directory 服务器：
  - 禁用详细日志记录
  - 禁用访问日志
  - 删除存档的日志文件



### 注意

目录服务器总是继续编写错误日志，即使达到阈值也是如此。

- 如果可用磁盘空间低于配置的阈值的一半，Directory 服务器会在定义的宽限期内关闭。
- 如果可用磁盘空间小于 4 KB，则目录服务器会立即关闭。

如果释放了磁盘空间，则 Directory Server 将中止关闭过程并重新启用之前禁用的所有日志设置。

### 2.4.1. 使用命令行配置本地磁盘监控

使用命令行配置本地磁盘监控：

1. 启用磁盘监控功能，设置阈值和宽限期：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-disk-monitoring=on nsslapd-disk-monitoring-threshold=3000000000 nsslapd-disk-monitoring-grace-period=60
```

这个命令将可用磁盘空间的阈值设置为 3 GB，宽限期设置为 60 秒。

- 另外，还可通过启用 **nsslapd-disk-monitoring-logging-critical** 参数来配置 Directory 服务器都禁用访问日志或删除归档的日志：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-disk-monitoring-logging-critical=on
```

- 重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

## 2.4.2. 使用 Web 控制台配置本地磁盘监控

使用 Web 控制台配置本地磁盘监控：

- 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的 [使用 Web 控制台登录到目录服务器](#) 一节。
- 选择实例。
- 打开 **Server Settings** 菜单，然后选择 **Server Configuration**。
- 启用 **磁盘空间监控**，并在分钟内设置阈值（以字节为单位）。

Server Configuration Settings	
Server Hostname	<input type="text" value="server.example.com"/>
Server Port	<input type="text" value="389"/>
Listen Host Address	<input type="text"/>
Database Backup Directory	<input type="text" value="/var/lib/dirsrv/slapd-instance_name/bak"/>
LDIF File Directory	<input type="text" value="/var/lib/dirsrv/slapd-instance_name/ldif"/>
<input checked="" type="checkbox"/> <b>Enable Disk Space Monitoring</b>	
Disk Monitoring Threshold	<input type="text" value="3221225472"/>
Disk Monitoring Grace Period	<input type="text" value="60"/>
<input type="checkbox"/> <b>Preserve Logs</b>	

这个示例将监控阈值设置为 3 GB (3,221、225、472 字节)，以及目录服务器在达到 **60** 分钟后关闭实例前的时间。

- （可选）配置 Directory 服务器都通过选择 **Preserve Logs** 来禁用访问日志或删除归档的日志。
- 单击 **Save Configuration**。
- 单击 **Actions** 按钮，然后选择 **Restart Instance**。

## 第 3 章 调整锁定数

锁定目录服务器中的机制控制目录服务器进程可以同时运行多少个副本。例如，在导入作业过程中，Directory 服务器在 `/run/lock/dirsrv/slaped-instance_name/imports/` 目录中设置一个锁定，以防止 `ns-slaped` (Directory Server) 进程、另一个导入或导出操作运行。

如果服务器没有可用的锁定，则会在 `/var/log/dirsrv/slaped-instance_name/errors` 文件中记录以下错误：

```
libdb: Lock table is out of available locks
```

如果错误消息显示锁定表没有可用锁定，请加倍锁定的数量。如果问题仍然存在，请再次加倍。

### 3.1. 手动监控锁定数

要使用命令行监控锁定数量，请输入：

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
-s sub -b "cn=database,cn=monitor,cn=ldb database,cn=plugins,cn=config"
nsslapd-db-current-locks nsslapd-db-max-locks
```

有关监控属性的详情，请查看 [目录服务器配置、命令和文件参考中的](#) 描述。

### 3.2. 通过监控自由数据库锁定来避免数据崩溃

不使用数据库锁定可能会导致数据崩溃。要避免这种情况，默认情况下 Directory 服务器会监控剩余的可用数据库数量，且主动数据库锁定数量等于或大于 90% 时，目录服务器会中止所有搜索。

您可以更改间隔和阈值：

- 例如，要将间隔设置为 **600** 毫秒，其阈值设为 **85** 百分比，请输入：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --locks-
monitoring-enabled on --locks-monitoring-pause 600 --locks-monitoring-threshold 85
```

`--locks-monitoring-enabled on` 选项确保已启用该功能。



#### 注意

如果您设置太高的间隔，服务器可以在下一次监控检查发生前耗尽锁定。设置太短的间隔可能会减慢服务器的速度。

- 重启实例：

```
# dsctl instance_name restart
```

### 3.3. 使用命令行设置锁定数

使用命令行设置锁定数量：

- 使用 `dsconf backend config set` 命令更新锁定数量。例如，要将值设为 **20000**：

■

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --locks=20000
```

2. 重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

### 3.4. 使用 WEB 控制台设置锁定数

使用 Web 控制台设置锁定数量：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 *Red Hat Directory Server Administration Guide* 中的 [使用 Web 控制台登录到 目录服务器](#) 一节。
2. 选择 实例。
3. 打开 **Database** 菜单，然后选择 **Global Database Configuration**。
4. 单击 **Show Advanced Settings**。
5. 更新 **Database Locks** 字段中的值。
6. 单击 **Save Configuration**。
7. 单击 **Actions** 按钮，然后选择 **Restart Instance**。

## 第 4 章 提高搜索性能（及平衡读取性能）

为目录改进搜索操作的最有效方法是，为条目配置全面的索引，以及对搜索结果的合理限制。

### 4.1. 使用索引

索引（如它表示）是一个标签，显示特定条目包含特定属性，而不必包含与条目相关的任何其他详情（这样可节省空间并更快返回搜索结果）。每个索引都围绕 Directory Server 属性进行组织，并以某种方式与该属性匹配：

- *存在索引（已弃用）* 只是显示哪个条目包含属性。
- *等性索引(eq)* 显示与特定搜索字符串匹配的属性值。
- *大约索引(approx)* 用于高效的语音搜索，显示具有可手机与字符串匹配的值的条目。
- *子字符串索引(sub)* 将属性值的任何子字符串与给定的搜索字符串匹配。（如果服务器维护非常昂贵，此索引就会非常昂贵。）
- *国际索引* 使用匹配规则来匹配目录中的字符串，其中包含英语以外的语言值。



#### 注意

Red Hat Directory Server Administration Guide 中的 [管理索引](#) 章节中包括了更详细的信息。

但是，只创建索引不会直接提高服务器性能。维护索引通过针对服务器维护的每个索引验证更改中的每个属性，从而在每次修改、添加和删除操作方面都造成负担：

1. 目录服务器收到一项添加或修改操作。
2. 目录服务器检查索引属性，以确定是否为属性值维护索引。
3. 如果创建的属性值被索引，则 Directory 服务器会生成新的索引条目。
4. 服务器完成后，会根据客户端请求创建实际属性值。

例如，Directory 服务器添加该条目：

```
dn: cn=John Doe, ou=People,dc=example,dc=com
objectclass: top
objectClass: person
objectClass: orgperson
objectClass: inetorgperson
cn: John Doe
cn: John
sn: Doe
ou: Manufacturing
ou: people
telephoneNumber: 408 555 8834
description: Manufacturing lead for the Z238 line of widgets.
```

Directory 服务器正在维护以下索引：

- **cn**（通用名称）和 **sn** (surname) 属性的等同性、大约和子字符串索引。

- 平板和子字符串索引，用于电话号码属性。
- description 属性的子字符串索引。

当在目录中添加该条目时，Directory 服务器必须执行以下步骤：

1. 为 **John** 和 **John Doe** 创建 **cn** 相等索引条目。
2. 为 **John** 和 **John Doe** 创建适当的 **cn** 大约索引条目。
3. 为 **John** 和 **John Doe** 创建适当的 **cn** 子字符串索引条目。
4. 为 **Doe** 创建 **sn** 相等索引条目。
5. 为 **Doe** 创建适当的 **sn** 大约索引条目。
6. 为 **Doe** 创建适当的 **sn** 子字符串索引条目。
7. 为 **408 555 8834** 创建电话号码相等的索引条目。
8. 为 **408 555 8834** 创建适当的电话号码子字符串索引条目。
9. 为 Manufacturing 创建适当的描述子字符串索引条目，导致小部件的 **Z238** 行。为字符串生成大量子字符串条目。

在创建新索引前，请确定平衡索引的开销与搜索性能的潜在改进。特别重要的是，请将您维护的索引类型与目录中存储的信息类型以及用户通常搜索的信息类型匹配。

- 对于通常包含数字（如电话号码）的属性而言，大约索引效率不高。
- 子字符串索引无法用于二进制属性。
- 如果值较大（例如旨在包含摄影或含有加密数据的密码的属性），则应避免使用相等的索引。
- 在搜索中，维护不使用的属性的索引会增加开销，而不会提高全局搜索性能。
- 没有索引的属性仍可在搜索请求中指定，但根据搜索类型，搜索性能可能会有很大降级。
- 您维护的索引越大，所需的磁盘空间就越高。



### 注意

创建索引对于具有高搜索操作负载和低修改操作负载的目录更有效。

## 4.2. 调优 DIRECTORY 服务器资源设置

您可以配置多个参数来管理并改进 Directory 服务器使用的资源数量。

### 4.2.1. 使用命令行更新目录服务器资源设置

使用命令行更新服务器资源设置：

1. 更新性能设置：

```
dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
parameter_name=setting
```

您可以设置以下参数：

- **nsslapd-threadnumber** : 设置 worker 线程数量。
- **nsslapd-maxdescriptors** : 设置文件描述符的最大数量。
- **nsslapd-timelimit** : 设置搜索时间限制。
- **nsslapd-sizelimit** : 设置搜索大小限制。
- **nsslapd-pagedsizelimit** : 设置页面搜索大小限制。
- **nsslapd-idletimeout** : 设置闲置连接超时。
- **nsslapd-ioblocktimeout** : 设置输入/输出(I/O)块超时。
- **nsslapd-ndn-cache-enabled** : 启用或禁用规范化 DN 缓存。
- **nsslapd-ndn-cache-max-size** : 如果启用了 **nsslapd-ndn-cache-size**, 则设置规范化 DN 缓存大小。
- **nsslapd-outbound-ldap-io-timeout** : 设置出站 I/O 超时。
- **nsslapd-maxbersize** : 设置最大基本编码规则(BER)大小。
- **nsslapd-maxsasliosize** : 设置最大简单身份验证和安全层(SASL) I/O 大小。
- **nsslapd-listen-backlog-size** : 设置接收进入连接的最大套接字数。
- **nsslapd-max-filter-nest-level** : 设置最大嵌套过滤器级别。
- **nsslapd-ignore-virtual-attrs** : 启用或禁用虚拟属性查找。
- **nsslapd-connection-nocanon** : 启用或禁用反向 DNS 查找。
- **nsslapd-enable-turbo-mode** : 启用或禁用 turbo 模式功能。

有关这些参数的详情，请查看 [红帽目录服务器配置、命令和文件参考](#) 中的描述。

## 2. 重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

### 4.2.2. 使用 Web 控制台更新目录服务器资源设置

使用 Web 控制台更新服务器资源设置：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 Red Hat [Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择实例。
3. 打开 Server Settings 菜单，然后选择 **Tuning & Limits**。
4. 更新设置。（可选）点击 **Show Advanced Settings** 显示所有设置。



### Server Tuning & Limits

Number Of Worker Threads	<input type="text" value="16"/>	Idle Connection Timeout	<input type="text" value="3600"/>
Maximum File Descriptors	<input type="text" value="1048576"/>	I/O Block Timeout	<input type="text" value="10000"/>
Search Time Limit	<input type="text" value="3600"/>	<input checked="" type="checkbox"/> Enable Normalized DN Cache	
Search Size Limit	<input type="text" value="2000"/>	Normalized DN Cache Size	<input type="text" value="20971520"/>
Paged Search Size Limit	<input type="text" value="0"/>		

The size of the normalized DN cache in bytes (nsslapd-ndn-cache-max-size).

▼ Hide Advanced Settings

Outbound IO Timeout	<input type="text" value="300000"/>	<input type="checkbox"/> Disable Virtual Attribute Lookups	
Maximum BER Size	<input type="text" value="2097152"/>	<input checked="" type="checkbox"/> Disable Reverse DNS Lookups	
Maximum SASL IO Size	<input type="text" value="2097152"/>	<input checked="" type="checkbox"/> Enable Connection Turbo Mode	
Listen Backlog Size	<input type="text" value="128"/>		
Maximum Nested Filter Level	<input type="text" value="40"/>		

要显示参数的 **cn=config** 条目中的工具提示和对应属性名称，请将鼠标光标悬停在设置上。详情请查看 [红帽目录服务器配置、命令和文件参考](#) 中的参数描述。

5. 单击 **Save Configuration**。
6. 单击 **Actions** 按钮，然后选择 **Restart Instance**。

## 4.3. 设置索引扫描限制

在大型目录中，搜索结果列表会变得非常大。具有一千个 **inetorgperson** 条目的目录将有一千个条目，它返回的过滤器（如 **(objectclass=inetorgperson)**），而 **sn** 属性的索引将至少有一个条目。

从数据库加载长 ID 列表可显著降低搜索性能。配置参数 **nsslapd-idlistscanlimit** 会对在键被视为与整个主索引（表示搜索被视为具有不同资源限值集）之前读取的 ID 数量设置限制。

对于大型索引，实际上，将与索引匹配的搜索视为未索引搜索时更有效。搜索操作只能查找处理结果（整个目录）的一个位置，而不要搜索几乎是目录大小的索引，加上目录本身。

**nsslapd-idlistscanlimit** 属性的默认值为 **4000**，它为常见的数据库大小和访问模式提供了良好的性能。通常不需要更改此值。如果数据库索引的小于 4000 条目，但仍然小于总体目录，则增加扫描限制可提高搜索，否则达到 4000 的默认限制。

另一方面，降低限制可能会显著提高搜索速度，否则按 4000 条目限制，但不需要扫描每个条目的位置。

### 4.3.1. 使用命令行设置索引扫描限制

使用命令行设置索引扫描限制：

1. 例如，要将目录服务器在搜索操作期间搜索的条目 ID 数量设置为 **8000**：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --idlistscanlimit=8000
```

2. 重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

### 4.3.2. 使用 Web 控制台设置索引扫描限制

使用 Web 控制台设置索引扫描限制：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 Red Hat [Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择实例。
3. 在 Database 选项卡上，选择 **Global Database Configuration**。
4. 更新 **ID List Scan Limit** 字段中的值。
5. 单击 **Save Configuration**。
6. 单击 **Actions** 按钮，然后选择 **Restart Instance**。

## 4.4. 精细约束 ID 列表大小

在大型数据库中，一些查询可能会消耗大量 CPU 和 RAM 资源。要提高性能，您可以使用 `nsldap-idlistscanlimit` 属性设置默认 ID 扫描限制，该限制应用到数据库中的所有索引。但是，在某些情况下，为特定索引定义限制或不使用 ID 列表非常有用。您可以使用 `nsIndexIDListScanLimit` 属性为不同类型的搜索过滤器设置单独的设置。

要设置限制，例如，对于 `objectClass` 属性，请将 `nsIndexIDListScanLimit` 参数添加到 DN `cn=objectclass,cn=index,cn=userRoot,cn=idbm database,cn=plugins,cn=config`。

`nsIndexIDListScanLimit` 属性是多值，将以下参数列表用作值：

```
nsIndexIDListScanLimit: limit=NNN [type=eq[,sub,...]] [flags=AND[,XXX,...]] [values=val[,val,...]]
```

- **limit**: ID 列表的最大大小。有效值为：
  - **-1**: 无限。
  - **0** : 不要使用索引。
  - **1 到最大 32 位整数(2147483647)**: 最大 ID 数。
- **键入**: 可选。索引的类型。**Eq**、**子**、**pre** 等。该值必须是为索引定义指定的实际 `nsIndexType` 之一。例如，如果没有定义 `nsIndexType=eq`，则无法使用 `type=eq`。
- **标志** : 可选。更改应用扫描限制行为的标记。有效值为：
  - **AND** : 仅应用扫描限制来搜索属性出现在 **AND** 子句中。
  - **或** : 仅应用扫描限制以搜索属性出现在 **OR** 子句中。
- **值** : 可选。要应用限制，以逗号分隔的值列表必须与搜索过滤器匹配。由于匹配一次是一次的，因此如果任何值匹配，值将会匹配。

值必须一次只用于一种类型。

值必须与索引类型对应，且必须与应用索引的属性的语法对应。例如，如果您指定了基于整数的属性 `uidNumber`，并且为 `eq` 索引，则无法使用 `type=eq values=abc`。

如果值包含需要转义的空格、逗号、NULL 或其他值，则应使用 LDAP 过滤器转义语法：反斜杠 (\)，后接该字符的 2 位数字代码。在以下示例中，DN 值中的逗号使用 \2C 转义。

```
nsIndexIDListScanLimit: limit=0 type=eq
values=uid=user\2Ccn=People\2Cdc=example\2Cdc=com
```

#### 例 4.1. Setting nsIndexIDListScanLimit

在包含对象类 `inetOrgPerson` 的大型数据库中，搜索 `(& (objectClass=inetOrgPerson) (uid=user))` 首先会创建一个与 `objectClass=inetOrgPerson` 匹配的所有 1,000 万 ID 的 ID 列表。当数据库应用过滤器的第二部分时，它将搜索结果列表与 `uid=user` 匹配。在这种情况下，定义特定索引的限制或全部不使用 ID 列表非常有用。

要设置没有为 **AND** 子句中的 `objectClass=inetOrgPerson` 创建 ID 列表，请添加以下 `nsIndexIDListScanLimit`：

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=objectclass,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config
changetype: modify
replace: nsIndexIDListScanLimit
nsIndexIDListScanLimit: limit=0 type=eq flags=AND values=inetOrgPerson

modifying entry "cn=objectclass,cn=index,cn=userRoot,cn=ldbm database,cn=plugins,cn=config"
```

在 **AND** 子句中使用 `nsIndexIDListScanLimit` 时，没有为 `objectClass=inetOrgPerson` 创建 ID 列表。在所有其他情况下，应用了 `nsIndexIDListScanLimit` 的值。

## 4.5. 为搜索调整数据库缓存

影响搜索性能的数据库属性主要定义供服务器使用的内存量。可以为数据库缓存大小属性设置的最大值取决于机器上的实际内存量。粗略地说，计算机上的可用内存量始终应大于默认数据库缓存大小的总和，以及每个条目缓存大小的总和。

在更改这些缓存大小属性时要小心。使用这些属性提高服务器性能取决于数据库的大小、机器上可用的物理内存以及目录搜索是否随机（也就是说，如果目录客户端正在搜索随机且广泛分散的目录数据）。

如果数据库不适合于内存，如果搜索是随机的，请尝试增加这些属性上设置的值不会帮助目录性能。实际上，更改这些属性可能会给整体性能造成负面影响。

用于存储目录数据的每个数据库的属性可以调整大小。

要改进搜索操作的缓存命中比率，请通过编辑 `nsIndexIDListScanLimit` 参数的值来增加目录服务器在数据库缓存中维护的数据量，如第 6.5 节“设置数据库缓存大小”所述。

## 4.6. 管理特殊条目

目录服务器将 `cn=config` 条目存储在 `/etc/dirsrv/slapd-instance_name/dse.ldif` 配置文件中，而不与常规条目位于同一个高度可扩展数据库中。因此，不要将常规用户或组存储在 `cn=config` 中。

## 第5章 调整事务日志记录

每个目录服务器包含一个事务日志，负责编写它所管理的所有数据库的操作。每当执行修改等目录数据库操作时，服务器都会为该LDAP操作调用的所有数据库操作创建一个数据库事务。这包括更新条目索引文件中的条目数据，并更新所有属性索引。如果所有操作都成功，服务器会提交事务，将操作写入事务日志，并验证整个事务是否已写入到磁盘。如果其中任何这些操作失败，服务器会回滚事务，并且丢弃所有操作。在服务器上，这种方法以原子方式保证更新操作是原子的。整个操作会永久成功，且不可避免，或者失败。

目录服务器（通过内部托管线程）将事务日志的内容刷新到实际数据库索引文件中，并检查事务日志是否需要修剪。

如果服务器遇到失败（如关机），并正常关闭，则有关最近目录更改的信息仍由事务日志保存。服务器重启时，目录会自动检测到错误条件，并使用数据库事务日志来恢复数据库。

虽然数据库事务日志和数据库恢复是无需干预的自动过程，但建议您调优一些数据库事务日志属性来优化性能。



### 警告

事务日志记录属性仅适用于系统修改和诊断。这些设置应该只在红帽技术支持的指导下修改。设置这些属性和其他配置属性不一致，可能会导致目录不稳定。

### 5.1. 将数据库目录移到9月磁盘或分区中

要实现更高的性能，在快速驱动器中存储目录服务器数据库和事务日志，如非易失性内存 express(NVMe)驱动器或SSD。

例如，如果您已在运行了目录服务器实例，并希望将 `/dev/nvme0n1p1` 分区挂载到 `/var/lib/dirsrv/slaped-instance_name/db/` 目录中：

1. 停止实例：

```
# systemctl stop dirsrv@instance_name
```

2. 将 `/dev/nvme0n1p1` 分区挂载到临时目录。例如：

```
# mount /dev/nvme0n1p1 /mnt/
```

3. 将 `/var/lib/dirsrv/slaped-instance_name/db/` 目录的内容复制到临时挂载点：

```
# mv /var/lib/dirsrv/slaped-instance_name/db/* /mnt/
```

4. 卸载临时目录：

```
# umount /mnt/
```

5. 如果 `/var/lib/dirsrv/slaped-instance_name/db/` 也是一个单独的挂载点，卸载该目录：

```
# umount /var/lib/dirsrv/slaped-instance_name/db/
```

- 更新 `/etc/fstab` 文件，以便在系统引导时自动将 `/dev/nvme0n1p1` 分区挂载到 `/var/lib/dirsrv/slaped-instance_name/db/`。详情请查看 [Red Hat System Administrator 指南中的对应部分](#)。

- 挂载文件系统。如果您将条目添加到 `/etc/fstab` 中：

```
# mount /var/lib/dirsrv/slaped-instance_name/db/
```

- 如果 SELinux 在 **enforcing** 模式下运行，请恢复 SELinux 上下文：

```
# restorecon -Rv /var/lib/dirsrv/slaped-instance_name/db/
```

- 启动实例：

```
# systemctl start dirsrv@instance_name
```

## 5.2. 更改数据库检查点 INTERVAL

在常规时间段内，Directory 服务器会将操作写入事务日志中的数据库索引文件，并在数据库事务日志中记录检查点条目。通过指示已将哪些更改写入数据库索引，检查点条目表示从事务日志开始恢复的位置，从而加快恢复过程。

默认情况下，Directory 服务器设置为每 60 秒向数据库事务日志发送检查点条目。增加检查点间隔可能会增加目录写入操作的性能。但是，增加检查点间隔可能会增加在关闭后恢复目录数据库所需的时间，并且由于数据库事务日志文件而需要更多磁盘空间。因此，如果您熟悉数据库优化，且可以完全评估更改的影响，才修改此属性。

### 5.2.1. 使用命令行更改数据库检查点 Interval

要使用命令行更改数据库检查点间隔，请输入：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --checkpoint-interval=120
```

这个示例将间隔更改为 120 秒。

### 5.2.2. 使用 Web 控制台更改数据库检查点 Interval

使用 Web 控制台更改数据库检查点间隔：

- 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 Red Hat [Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
- 选择实例。
- 在 Database 选项卡上，选择 **Global Database Configuration**。
- 单击 **Show Advanced Settings**。
- 更新 **Database Checkpoint Interval** 字段中的值。

- 单击 **Save Configuration**。

### 5.3. 禁用 DURABLE TRANSACTIONS

持久的事务日志记录意味着每个LDAP 更新操作（由事务中的一系列数据库操作组成）是物理写入磁盘。尽管每个LDAP 操作都可以由多个数据库操作组成，但每个LDAP 操作都被视为一个数据库事务。每个LDAP 操作既是 atomic 和 durable。



#### 警告

关闭 durable 事务可让您提高 Directory Server 的写入性能，而出现数据丢失的风险。

禁用持久的事务日志记录时，每个目录数据库操作都会写入数据库事务日志文件，但可能不会立即实际写入磁盘。如果目录更改被写入逻辑数据库事务日志文件，但没有在系统崩溃时物理写入磁盘，则无法恢复更改。禁用持久事务后，恢复的数据库是一致的，但不反映在系统崩溃之前完成的任何LDAP 写入操作的结果。

默认情况下启用持久的数据库事务日志记录。禁用持久的事务日志：

1. 停止 Directory 服务器实例：

```
# dsctl instance_name stop
```

2. 编辑 `/etc/dirsrv/slapd-instance_name/dse.ldif` 文件，并将 `cn=config,cn=ldbm database,cn=plugins,cn=config` 条目中的 `nsslapd-db-durable-transaction` 参数设置为 `off`：

```
dn: cn=config,cn=ldbm database,cn=plugins,cn=config
...
nsslapd-db-durable-transaction: off
...
```

3. 启动 Directory 服务器实例：

```
# dsctl instance_name start
```

### 5.4. 指定事务批处理

要在不需要完全事务持久性时提高更新性能，请使用以下命令：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --txn-batch-val=value
```

`--txn-batch-val` 指定在目录服务器将其提交到事务日志前批处理的事务数量。将此值设置为大于 0 的值会导致服务器延迟提交事务，直到排队的事务数量等于这个值。

## 第 6 章 管理数据库缓存设置

目录服务器使用以下缓存：

- Entry cache，其中包含单个目录条目。
- DN 缓存 用于将 DN 和 RDN 与条目相关联。
- 数据库缓存，其中包含数据库索引文件 192.168.1.0/24 **db** 和 **the db4** 文件。

为了提高性能，所有缓存大小都必须能够存储其所有记录。如果不使用推荐的自动大小功能，且没有足够可用的 RAM，请在前面显示的顺序为缓存分配可用内存。

### 6.1. DATABASE 和 ENTRY CACHE AUTO-SIZING 功能

默认情况下，Directory 服务器会自动决定数据库和条目缓存的最佳大小。在实例启动时，自动大小会根据服务器的硬件资源优化这两个缓存的大小。



#### 重要

红帽建议使用自动调整设置。不要手动设置条目缓存大小。

#### 6.1.1. 手动重新启用数据库和条目缓存自动大小

如果您从 10.1.1 之前的版本升级实例，或者之前手动设置条目缓存大小，您可以为条目缓存启用自动调整。

**cn=config,cn=ldbm database,cn=plugins,cn=config** 条目中的以下参数控制自动大小：

#### **nsslapd-cache-autosize**

这个设置控制是否为数据库和条目缓存启用自动大小。自动大小被启用：

- 对于数据库和条目缓存，如果 **nsslapd-cache-autosize** 参数设置为大于 0 的值。
- 对于数据库缓存，如果 **nsslapd-cache-autosize** 和 **nsslapd-dbcachesize** 参数设置为 0。
- 对于条目缓存，如果 **nsslapd-cache-autosize** 和 **nsslapd-cachememsize** 参数设置为 0。

#### **nsslapd-cache-autosize-split**

该值设定用于数据库缓存的 RAM 百分比。剩余百分比用于条目缓存。

将 1.5 GB RAM 用于数据库缓存不会提高性能。因此，Directory 服务器会限制数据库缓存 1.5 GB。

启用数据库和条目缓存自动大小：

1. 停止 Directory 服务器实例：

```
# systemctl stop dirsrv@instance_name
```

2. 备份 `/etc/dirsrv/slapd-instance_name/dse.ldif` 文件：

```
# cp /etc/dirsrv/slapd-instance_name/dse.ldif \
/etc/dirsrv/slapd-instance_name/dse.ldif.bak.$(date "+%F_%H-%M-%S")
```

3. 编辑 `/etc/dirsrv/slapped-instance_name/dse.ldif` 文件：

- a. 设置用于数据库和条目缓存的可用系统 RAM 百分比。例如，设置 10%：

```
nsslapd-cache-autosize: 10
```



### 注意

如果将 `nsslapd-cache-autosize` 参数设置为 0，还必须设置：

- `cn=config,cn=ldb database,cn=plugins,cn=config` 条目中的 `nsslapd-dbcachesize` 到 0，以启用 auto-sized 数据库缓存。
- `cn=database_name,cn=ldb database,cn=plugins,cn=config` 条目中的 `nsslapd-cachememsize` 到 0，以便为数据库启用自动条目缓存。

- b. 另外，还可为数据库缓存设置可用系统 RAM 中所使用的百分比。例如，设置 40%：

```
nsslapd-cache-autosize-split: 40
```

目录服务器将剩余的 60% 的可用内存用于条目缓存。

- c. 保存更改。

4. 启动 Directory 服务器实例：

```
# systemctl start dirsrv@instance_name
```

## 例 6.1. `nsslapd-cache-autosize` 和 `nsslapd-cache-autosize-split` Parameter

以下设置是这两个参数的默认设置：

```
nsslapd-cache-autosize: 10
nsslapd-cache-autosize-split: 40
```

使用这些设置时，将使用系统的可用 RAM 的 10% (`nsslapd-cache-autosize`)。在这个内存中，40% 用于数据库缓存 (`nsslapd-cache-autosize-split`)，剩余的 60% 用于条目缓存。

根据可用 RAM，这会导致以下缓存大小：

可用 RAM 的 GB	数据库缓存大小	条目缓存大小
1 GB	40 MB	62 MB
2 GB	82 MB	122 MB
4 GB	164 MB	245 MB
8 GB	328 MB	492 MB



可用 RAM 的 GB	数据库缓存大小	条目缓存大小
16 GB	512 MB <sup>[a]</sup>	1,126 MB
32 GB	512 MB <sup>[a]</sup>	2,764 MB
64 GB	512 MB <sup>[a]</sup>	6,042 MB
128 GB	512 MB <sup>[a]</sup>	12,596 MB

[a] 目录服务器为 **nsslapd-dbcachesize** 参数应用 512 MB 限制。

## 6.2. 确定所需的缓存大小

**dsconf monitor dbmon** 命令可让您在运行时监控缓存统计信息。

要显示统计信息，请输入：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com monitor dbmon
DB Monitor Report: 2020-06-24 11:31:27
-----
Database Cache:
- Cache Hit Ratio: 50%
- Free Space: 397.31 KB
- Free Percentage: 2.2%
- RO Page Drops: 0
- Pages In: 2934772
- Pages Out: 219075

Normalized DN Cache:
- Cache Hit Ratio: 60%
- Free Space: 19.98 MB
- Free Percentage: 99.9%
- DN Count: 100000
- Evictions: 9282348

Backends:
- dc=example,dc=com (userroot):
  - Entry Cache Hit Ratio: 66%
  - Entry Cache Count: 50000
  - Entry Cache Free Space: 2.0 KB
  - Entry Cache Free Percentage: 0.8%
  - Entry Cache Average Size: 8.9 KB
  - DN Cache Hit Ratio: 21%
  - DN Cache Count: 100000
  - DN Cache Free Space: 4.29 MB
  - DN Cache Free Percentage: 69.8%
  - DN Cache Average Size: 130.0 B
```

(可选) 将 **-b back\_end** 或 **-x** 选项传递给命令，以显示特定后端或索引的统计信息。

如果您的缓存的大小足够，则 **DN Cache Count** 中的数字与 **Cache Count** 后端条目中的值匹配。另外，如果所有条目和 DN 都适合其相应的缓存，则 **Entry Cache Count** 的值与 **DN Cache Count** 值匹配。

示例输出显示：

- 只有 2.2% 的空闲数据库缓存才会保留：

```
Database Cache:
...
- Free Space:      397.31 KB
- Free Percentage: 2.2%
```

但是，为了有效地操作，至少需要 15% 免费数据库缓存。要确定数据库缓存的最佳大小，请计算 `/var/lib/dirsrv/slapd-instance_name/db/` 目录（包括子目录和 changelog 数据库）中所有，并将 12% 用于开销。

要设置数据库缓存，请参阅 [第 6.5 节“设置数据库缓存大小”](#)。

- **userroot** 数据库的 DN 缓存是良好的选择：

```
Backends:
- dc=example,dc=com (userroot):
...
- DN Cache Count:      100000
- DN Cache Free Space: 4.29 MB
- DN Cache Free Percentage: 69.8%
- DN Cache Average Size: 130.0 B
```

数据库的 DN 缓存包含 100000 记录，即缓存的 69,8% 可用，内存中的每个 DN 平均需要 130 字节。

要设置 DN 缓存，请参阅 [第 6.4 节“设置 DN 缓存的大小”](#)。

- **userroot** 数据库条目缓存的统计信息表示应提高条目缓存值以提高性能：

```
Backends:
- dc=example,dc=com (userroot):
...
- Entry Cache Count:      50000
- Entry Cache Free Space: 2.0 KB
- Entry Cache Free Percentage: 0.8%
- Entry Cache Average Size: 8.9 KB
```

条目缓存包含在这个数据库 50000 记录中，仅保留有 2 个千字节可用空间。要启用目录服务器缓存所有 100000 DN，缓存必须增加到最小 890 MB (100000 DN \* 8,9 KB 平均条目大小)。但是，红帽建议将最低要求大小调整到下一个最高 GB 并加倍。在这个示例中，条目缓存应设置为 2 千兆字节。

要设置条目缓存，请参阅第 6.3 节“手动设置 Entry Cache Size”。

### 6.3. 手动设置 ENTRY CACHE SIZE

条目缓存用于存储搜索和读取操作中使用的目录条目。将条目缓存设置为可让 Directory 服务器存储所有记录的最高性能影响。

如果没有配置条目缓存，Directory 服务器会从 id2entry.db 数据库文件读取条目，并将 DN 从磁盘格式转换为内存格式。存储在缓存中的条目可让服务器跳过磁盘 I/O 和转换步骤。



#### 注意

红帽建议自动根据硬件资源优化设置，而不是手动设置条目缓存大小。详情请查看第 6.1.1 节“手动重新启用数据库和条目缓存自动大小”。

#### 6.3.1. 使用命令行手动设置条目缓存大小

使用命令行手动设置条目缓存大小：

1. 禁用自动缓存调整：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --cache-autosize=0
```

2. 显示后缀及其对应的后端：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com suffix list
dc=example,dc=com (userroot)
```

这个命令显示每个后缀旁的后端数据库名称。下一步需要后缀的数据库名称。

3. 为数据库设置条目缓存大小：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix set --cache-memsize=2147483648 userRoot
```

此命令将条目缓存设置为 **2GB**。

4. 重启 **Directory Service** 实例：

```
# dsctl instance_name restart
```

### 6.3.2. 使用 Web 控制台手动设置 Entry Cache Size

使用 Web 控制台手动设置条目缓存大小：

1. 在 web 控制台中打开 **Directory Server** 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择实例。
3. 在 **Database** 选项卡上，选择 **Global Database Configuration**。
4. 禁用自动 缓存调优。
5. 单击 **Save Configuration**。
6. 单击 **Actions** 按钮，然后选择 **Restart Instance**。
7. 在 **Entry Cache Size (bytes)** 字段中设置数据库缓存的大小。
8. 单击 **Save Configuration**。

9. 单击 **Actions** 按钮，然后选择 **Restart Instance**。

## 6.4. 设置 DN 缓存的大小

**entryrdn** 索引用于将 DN 和 RDNs 与条目关联。它能让服务器有效地执行子树重命名、条目 **move**、和 **moddn** 操作。DN 缓存用于缓存 **entryrdn** 索引的内存中表示，以避免昂贵的文件 I/O 和转换操作。为获得最佳性能，特别是对于条目重命名和移动操作的限制，将 DN 缓存设置为可让目录服务器缓存数据库中的所有 DN 的大小。

如果 DN 没有存储在缓存中，Directory 服务器会从 **entryrdn.db** 索引数据库文件读取 DN，并将 DN 从磁盘格式转换为内存格式。存储在缓存中的 DNS 可让服务器跳过磁盘 I/O 和转换步骤。

### 6.4.1. 使用命令行设置 DN 缓存的大小

使用命令行设置数据库的 DN 缓存大小：

1. 显示后缀及其对应的后端：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com suffix list  
dc=example,dc=com (userroot)
```

这个命令显示每个后缀旁的后端数据库名称。下一步需要后缀的数据库名称。

2. 使用以下命令设定 DN 缓存大小：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix set --  
dncache-memsize=20971520 userRoot
```

此命令将 **userRoot** 数据库的 DN 缓存设置为 **20MB**。

3. 重启 **Directory Service** 实例：

```
# dsctl instance_name restart
```

### 6.4.2. 使用 Web 控制台设置 DN 缓存的大小

使用 Web 控制台设置数据库的 DN 缓存大小：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择实例。
3. 在 Database 选项卡中，选择要为其设置 DN 缓存大小的后缀。
4. 在 DN Cache Size (bytes) 字段中输入大小（以字节为单位）。
5. 单击 Save Configuration。
6. 单击 Actions 按钮，然后选择 Restart Instance。

## 6.5. 设置数据库缓存大小

数据库缓存包含数据库的 Berkeley 数据库索引文件，这意味着所有用于数据库属性索引的 192.168.1.0/24 db 和其他文件。这个值被传递给 Berkeley DB API 功能 `set_cachesize ()`。

这个缓存大小对 Directory 服务器性能的影响低于条目缓存大小，但如果设定了条目缓存大小后有可用的 RAM，增加分配给数据库缓存的内存量。

操作系统还具有文件系统缓存，与 RAM 使用量的数据库缓存竞争。请参阅操作系统文档，以查找有关文件系统缓存设置和监控文件系统缓存的信息。



### 注意

红帽建议自动根据硬件资源优化设置，而不是手动设置条目缓存大小。详情请查看 [第 6.1.1 节“手动重新启用数据库和条目缓存自动大小”](#)。

### 6.5.1. 使用命令行手动设置数据库缓存大小

使用命令行手动设置数据库缓存大小：

1. **禁用自动缓存调整，并设置数据库缓存大小：**

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --cache-autosize=0 --dbcachesize=268435456
```

此命令将数据库缓存设置为 **256MB**。

2. **重启 Directory Service 实例：**

```
# dsctl instance_name restart
```

### 6.5.2. 使用 Web 控制台手动设置数据库缓存大小

使用 Web 控制台手动设置数据库缓存大小：

1. **在 web 控制台中打开 Directory Server 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。**
2. **选择 实例。**
3. **在 Database 选项卡上，选择 Global Database Configuration。**
4. **禁用自动 缓存调优。**
5. **单击 Save Configuration。**
6. **将 Database Cache Size (bytes) 字段设置为数据库缓存大小。**
7. **单击 Save Configuration。**

8. 单击 **Actions** 按钮，然后选择 **Restart Instance**。

### 6.5.3. 将数据库缓存存储在 RAM 磁盘上

如果运行 **Directory** 服务器实例的系统有足够的可用 **RAM**，则可选择性地将数据库缓存存储在 **RAM** 磁盘上，以进一步提高性能：

1. 在 **RAM** 磁盘上为数据库缓存和元数据创建一个目录：

```
# mkdir -p /dev/shm/slaped-instance_name/
```

2. 在目录中设置以下权限：

```
# chown dirsrv:dirsrv /dev/shm/slaped-instance_name/  
# chmod 770 /dev/shm/slaped-instance_name/
```

3. 停止 **Directory** 服务器实例：

```
# systemctl stop dirsrv@instance_name
```

4. 编辑 `/etc/dirsrv/slaped-instance_name/dse.ldif` 文件，并在 `cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config` 条目中设置 `nsslaped-db-home-directory` 属性中的新路径：

```
dn: cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config  
...  
nsslaped-db-home-directory: /dev/shm/slaped-instance_name/
```

如果 `nsslaped-db-home-directory` 属性不存在，则使用新值将其添加到 `cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config` 条目。

5. 启动 **Directory** 服务器实例：

```
# systemctl start dirsrv@instance_name
```



**注意**

当数据库缓存存储在 RAM 磁盘上时，Directory 服务器需要在每次重启后重新创建它。因此，在重新创建缓存前，服务启动和初始操作会较慢。

## 第 7 章 设置目录服务器线程的数量

目录服务器用来同时处理连接数量会影响服务器的性能。例如，如果所有线程都忙于处理耗时的任务（如添加操作），新的传入连接会被排队，直到空闲线程可以处理请求。

如果服务器提供较少的 CPU 线程，配置更多线程数量可能会提高性能。但是，在有多个 CPU 线程的服务器上，设置太高的值不会进一步提高性能。

默认情况下，Directory 服务器会自动计算线程数量。此数值基于实例启动时服务器的硬件资源。



### 注意

红帽建议使用自动调整设置。不要手动设置线程数量。

### 7.1. 自动线程调优

如果您启用自动线程调整，Directory 服务器将使用以下优化的线程数量：

CPU 线程数	Directory Server Threads 的数量
1	16
2	16
4	24
8	32
16	48
32	64
64	96
128	192
256	384

CPU 线程数	Directory Server Threads 的数量
512	512 [a]
1024	512 [a]
2048	512 [a]
[a] 应用推荐的最大线程数量。	

### 7.1.1. 使用命令行启用自动线程调优

目录服务器可以根据可用的硬件线程自动设置线程数量。启用此功能：

1. 启用自动设置线程数量：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-threadnumber="-1"
```

2. 重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

#### 重要

如果您启用了线程数的自动设置，`nsslapd-threadnumber` 参数会在 Directory Server 运行时显示计算的线程数量。

### 7.1.2. 使用 Web 控制台启用自动线程调整

目录服务器可以根据可用的硬件线程自动设置线程数量。启用此功能：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到目录服务器一节。

2. 选择实例。
3. 打开 **Server Settings** 菜单，然后选择 **Tuning & Limits**。
4. 将 **Number Of Worker Threads** 字段设置为 **-1**。
5. 点击 **Save**。
6. 单击 **Actions** 按钮，然后选择 **Restart Instance**。



### 重要

如果您启用了自动设置，则 **Number Of Worker Threads** 字段会在 **Directory Server** 运行时显示计算的线程数量。

## 7.2. 手动设置线程数

在某些情况下，可能需要手动设置固定数量的目录服务器线程，而不是使用自动线程调整。



### 注意

例如，如果硬件线程数量发生变化，因为您增加了运行 **Directory** 服务器实例的虚拟机的 **CPU** 内核，您必须手动更新线程数量。有关使用优化和自动设置的详情，请参考 [第 7.1 节“自动线程调优”](#)。

### 7.2.1. 使用命令行手动设置线程数

使用命令行手动设置线程数量：

1. 设置线程数量：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-threadnumber="64"
```

此命令将线程数量设置为 64。

2. **重启 Directory 服务器实例：**

```
# dsctl instance_name restart
```

### 7.2.2. 使用 Web 控制台手动设置线程数量

使用 Web 控制台手动设置线程数量：

1. **在 web 控制台中打开 Directory Server 用户界面。详情请参阅 Red Hat [Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。**
2. **选择 实例。**
3. **打开 Server Settings 菜单，然后选择 Tuning & Limits。**
4. **将 Number Of Worker Threads 字段设置为线程数量。**
5. **点击 Save。**
6. **单击 Actions 按钮，然后选择 Restart Instance。**

## 第 8 章 调优复制性能

### 8.1. 提高多层次复制效率

在多分配复制环境中复制延迟，特别是在服务器使用广域网(WAN)连接时，当多个供应商同时接收更新时，可能很高。当一个供应商仅访问一个副本时，不会长时间释放副本。在这种情况下，其他供应商无法向这个消费者发送更新，这会增加复制延迟。

要在固定时间后发布副本，请在复制供应商和 hub 中设置 `nsds5ReplicaReleaseTimeout` 参数。



#### 注意

60 秒的默认值对于大多数环境来说是理想的选择。设定太大或太低的值可能会对复制性能造成负面影响。如果值设置太低，复制服务器会持续重新分配另一个服务器，并且服务器无法发送多个更新。在高流量复制环境中，更长的超时可以改进供应商独有访问副本的情况。然而，在大多数情况下，超过 120 秒的值会减慢复制速度。

#### 8.1.1. 使用命令行设置复制发行超时

使用命令行设置复制发行超时：

1. 设置超时值：

```
# dsconf -D "cn=Directory Manager" ldap://supplier.example.com replication set --suffix="dc=example,dc=com" --repl-release-timeout=70
```

此命令将 `dc=example,dc=com` 后缀的复制发行版本超时值设置为 70 秒。

2. 重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

#### 8.1.2. 使用 Web 控制台设置复制发行超时

使用 Web 控制台设置复制发行超时：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 Red Hat [Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择 实例。
3. 打开 Replication 菜单，然后选择 Configuration。
4. 单击 Show Advanced Settings。
5. 在 Replication Release Timeout 字段中设置超时值。
6. 单击 Save。
7. 单击 Actions 按钮，然后选择 Restart Instance。

## 第 9 章 调优数据库链路性能

通过更改 Directory 服务器的连接和线程管理，可以改进数据库链接性能。

### 9.1. 管理到远程服务器的连接

每个数据库链接维护一个到远程服务器的连接池。这部分论述了如何优化它们。

#### 9.1.1. 使用命令行管理与远程服务器的连接

本节论述了如何更新特定数据库的设置以及默认设置。

##### 9.1.1.1. 更新特定数据库的数据库链接连接管理设置

为特定数据库更新数据库连接管理设置：

1.

使用以下命令为数据库链接更新设置：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com chaining link-set  
parameter=value link_name
```

有关您可以设置的参数列表，请输入：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com chaining link-set --help
```

2.

重启 Directory 服务器实例：

```
# dsctl instance_name restart
```

##### 9.1.1.2. 更新默认数据库链路连接管理设置

要更新默认数据库连接管理设置，请使用以下命令：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com chaining config-set-def  
parameter=value
```



有关您可以设置的参数列表，请输入：

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com chaining config-set-def --help
```

### 9.1.2. 使用 Web 控制台管理到远程服务器的连接

本节论述了如何更新特定数据库的设置以及默认设置。

#### 9.1.2.1. 更新特定数据库的数据库链接连接管理设置

为特定数据库更新数据库连接管理设置：

1. 在 web 控制台中打开 Directory Server 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到 目录服务器一节。
2. 选择 实例。
3. 在 Database 选项卡中，选择您要更新的数据库链路配置。
4. 单击 **Show Advanced Settings**。
5. 更新高级设置区域中的字段：

要显示参数的 `cn=config` 条目中的工具提示和对应属性名称，请将鼠标光标悬停在设置上。详情请查看 [Red Hat Directory Server Configuration、命令和文件参考中的参数描述](#)。

6. 单击 **Save Configuration**。
7. 单击 **Actions** 按钮，然后选择 **Restart Instance**。

### 9.1.2.2. 更新默认数据库链路连接管理设置

更新默认数据库连接管理设置：

1. 在 web 控制台中打开 **Directory Server** 用户界面。详情请参阅 [Red Hat Directory Server Administration Guide](#) 中的使用 Web 控制台登录到目录服务器一节。
2. 选择实例。
3. 在 **Database** 选项卡上，选择 **Chaining Configuration**。

4.

更新 **Default Database Link Creation Settings** 区域中的字段：

Default Database Link Creation Settings	
Size Limit	<input type="text" value="2000"/>
Max Operations Per Conn	<input type="text" value="2"/>
Time Limit	<input type="text" value="3600"/>
Connection Lifetime	<input type="text" value="0"/>
Max TCP Connections	<input type="text" value="3"/>
Abandoned Op Check Interval	<input type="text" value="1"/>
Max LDAP Connections	<input type="text" value="20"/>
Abandoned Op Check Interval	<input type="text" value="1"/>
Max Binds Per Connection	<input type="text" value="20"/>
Database Link Hop Limit	<input type="text" value="10"/>
Bind Timeout	<input type="text" value="15"/>
Bind Retry Limit	<input type="text" value="3"/>
Check Local ACLs	<input type="checkbox"/>
Send Referral On Scoped Search	<input type="checkbox"/>
Allow Proxied Authentication	<input checked="" type="checkbox"/>
Use StartTLS	<input type="checkbox"/>

要显示参数的 `cn=config` 条目中的工具提示和对应属性名称，请将鼠标光标悬停在设置上。详情请查看 [Red Hat Directory Server Configuration](#)、[命令和文件参考](#) 中的参数描述。

5.

单击 **Save Default Settings**。

6.

单击 **Actions** 按钮，然后选择 **Restart Instance**。

## 9.2. 在正常处理过程中检测错误

通过检测数据库链接和远程服务器之间的正常链操作期间检测服务器性能，从而保护服务器性能。数据库链接有两个属性 - `nsMaxResponseDelay` 和 `nsMaxTestResponseDelay` - 协同工作来确定远程服务

器是否不再响应。

第一个属性 `nsMaxResponseDelay` 设置一个最长持续时间，以便 LDAP 操作完成。如果操作所需的时间超过此属性中指定的时间，则数据库链接的服务器会怀疑远程服务器不再在线。

满足 `nsMaxResponseDelay` 周期后，数据库链接会 ping 远程服务器。在 ping 中，数据库链接会发出另一个 LDAP 请求，即远程服务器中不存在的对象的简单搜索请求。ping 的持续时间使用 `nsMaxTestResponseDelay` 进行设置。

如果远程服务器在 `nsMaxTestResponseDelay` 周期通过前没有响应，则返回错误，连接将标记为 down。数据库链接和远程服务器之间的所有连接都将在 30 秒内被阻止，从而防止服务器性能下降。30 秒后，数据库链接向远程服务器发出的操作请求会正常继续。

这两个属性都存储在 `cn=config,cn=chaining 数据库,cn=plugins,cn=config` 条目中。下表描述了更详细的属性：

表 9.1. 数据库链路处理错误检测参数

属性名称	描述
<code>nsMaxResponseDelay</code>	在怀疑错误之前，能够让远程服务器响应数据库链接提出的 LDAP 操作请求的最大时间。这个周期以秒为单位指定。默认延迟周期为 <b>60</b> 秒。满足这个延迟周期后，数据库链接会测试与远程服务器的连接。
<code>nsMaxTestResponseDelay</code>	数据库链接发布的测试持续时间，以检查远程服务器是否响应。如果在此周期通过前没有返回来自远程服务器的响应，数据库链接会假定远程服务器停机，并且连接不会用于后续的操作。这个周期以秒为单位指定。默认测试响应延迟周期为 <b>15</b> 秒。

## 第 10 章 提高导入性能

大量条目大小或大量条目在导入操作过程中会对服务器性能造成负面影响。这部分论述了如何调整目录服务器设置和操作系统设置来提高导入性能。

### 10.1. 为大型数据库导入和利用大量属性导入来调优目录服务器

在以下情况下更新条目缓存：

- 您需要导入非常大的数据库。
- 您需要导入具有大属性的数据库，如存储证书链或镜像的二进制属性。

有关设置条目缓存大小的详情，请参考 [第 6.1 节 “Database 和 Entry Cache Auto-Sizing 功能”](#) 和 [第 6.3 节 “手动设置 Entry Cache Size”](#)。

### 10.2. 调优目录服务器以导入大量条目

当您导入大量条目时，最多用户进程数的操作系统设置可以限制目录服务器的性能。

- 要临时增加最大进程数，请输入：

```
# ulimit -u 32000
```

当用户注销时，更改将返回到默认设置。

- 要永久增加进程的最大数量，请参阅 [“如何设置 ulimit 值”](#)。

**附录 A. 修订历史记录**

请注意，修订号与该手册版本相关，而不是与 Red Hat Directory Server 的版本号相关。

**修订 11.5-1**                      **Tue May 10 2022**                      **Marc Muehlfeld**

本指南的 Red Hat Directory Server 11.5 发行版本。

**修订 11.4-1**                      **Tue Nov 09 2021**                      **Marc Muehlfeld**

本指南的 Red Hat Directory Server 11.4 发行版本。

**修订 11.3-1**                      **Tue May 11 2021**                      **Marc Muehlfeld**

本指南的 Red Hat Directory Server 11.3 发行版本。

**修订 11.2-1**                      **Tue Nov 03 2020**                      **Marc Muehlfeld**

本指南的 Red Hat Directory Server 11.2 发行版本。

**修订 11.1-1**                      **Tue Apr 28 2020**                      **Marc Muehlfeld**

本指南的 Red Hat Directory Server 11.1 发行版本。

**修订 11.0-1**                      **Tue Nov 05 2019**                      **Marc Muehlfeld**

本指南的 *Red Hat Directory Server 11.0* 发行版本。