Red Hat Enterprise Linux 9

Identity Management でのパフォーマンスの調整

パフォーマンスを向上させるための IdM サービス (Directory Server、KDC、SSSD など) の最適化

Last Updated: 2024-08-06

Red Hat Enterprise Linux 9 Identity Management でのパフォーマンスの調整

パフォーマンスを向上させるための IdM サービス (Directory Server、KDC、SSSD など) の最適化

法律上の通知

概要

Red Hat は、ほとんどのデプロイメントで適切に動作するように Identity Management (IdM) を調整します。ただし、特定のシナリオでは、レプリケーションアグリーメント、Directory Server、Kerberos Key Distribution Center (KDC)、または System Security Services Daemon (SSSD) などの IdM コンポーネントを調整すると有益な場合があります。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

Jira からのフィードバック送信 (アカウントが必要)

1. Jira の Web サイトにログインします。
2. 上部のナビゲーションバーで **Create** をクリックします。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. ダイアログの下部にある **Create** をクリックします。

第1章 IDM のチューニングにおける重要な考慮事項

Identity Management のコンポーネントサービスは、ほとんどのデプロイメントで最適に機能するように調整されています。システム管理者は、特定の環境でのニーズに合わせて、IdM サービスのパフォーマンスを調整できます。

重要な留意事項

- IdM デプロイメントはそれぞれ、ハードウェア、ソフトウェア、ネットワーク、データ、ワークロード、およびその他の要因を一意に組み合わせます。ある環境で調整が有効であっても、別の環境では悪影響がある場合があります。

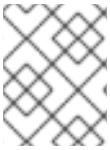
- パフォーマンスの調整は、反復的で実験的なプロセスです。Red Hat は、1 回に調整する変数は 1 つだけにして、ご使用の環境への影響を監視することを推奨します。変数 1 つの調整で目的とする結果が得られた場合には、これまでの調整によるパフォーマンスの監視を継続しつつ、次の変数を調整します。

第2章 ハードウェア推奨事項

ハードウェアでは、RAM の容量を適切に確保することが最も重要になります。システムに十分な RAM があるようにしてください。一般的な RAM の要件は次のとおりです。

- 10,000 ユーザーおよび 100 グループには、最低 4 GB の RAM と 4 GB のスワップ領域を割り当てます。
- 100,000 ユーザーおよび 50,000 グループには、最低 16 GB の RAM と 4 GB のスワップ領域を割り当てます。

大規模なデプロイメントでは、データのほとんどがキャッシュに保存されるため、ディスクスペースを増やすよりも RAM を増やす方が効果的です。通常、メモリーを増やすと、キャッシュ機能により、サイズが大きいデプロイメントでパフォーマンスが改善されます。



注記

基本的なユーザーエントリーまたは証明書のあるシンプルなホストエントリーのサイズは約 5 ~ 10 KB になります。

第3章 IDM サーバーのパフォーマンスに関する推奨事項

次の表に、IdM 環境に同時に追加または登録できるユーザーおよびクライアントの最大数を示します。これ以下の値を使用することで、IdM サーバーのパフォーマンスが安定します。

表3.1 IdM サーバーごとの最大数

アクション	説明	数値
クライアントの登録	登録に失敗することなく、IdM サーバーに同時に登録できる IdM クライアントの最大数。	130
ユーザーの追加	ユーザーの追加に失敗することなく、**ipa user-add[]** コマンドを使用してさまざまな IdM クライアントから同時に追加できるユーザーの最大数。 IdM API **batch** コマンドを使用すると、同じ時間でさらに多くのユーザーを追加できます。ユーザーは 100 ユーザー単位で追加することを推奨します。**batch** コマンドの詳細は、バッチを使用した IdM API コマンドの実行 を参照してください。	325
クライアントの認証	認証に失敗することなく、同時に認証できる IdM クライアントの最大数。	800
ユーザーグループへのメンバーの追加	グループに新しいメンバーを追加するための時間を超過することなく追加できる、グループのメンバーの推奨数。IdM には、メンバーをグループに追加するための通常の時間枠を 2 秒とするルールがあります。さらにメンバーを追加することもできますが、アクションの時間は徐々に延長されます。	1500

第4章 IDM でのフェイルオーバー、負荷分散、高可用性

Identity Management (IdM) には、IdM クライアント向けのフェイルオーバーメカニズムと、IdM サーバー向けの負荷分散および高可用性機能があります。

4.1. クライアント側のフェイルオーバー機能

- デフォルトでは、IdM クライアントの **SSSD** サービスは、DNS からのサービス (SRV) リソースレコードを使用して、接続先に最も適した IdM サーバーを自動的に決定するように設定されています。この動作は、**/etc/sssd/sssd.conf** ファイルの **ipa_server** パラメーターの **_srv_** オプションで制御します。

```
[root@client ~]# cat /etc/sssd/sssd.conf

[domain/example.com]
id_provider = ipa
ipa_server = _srv_, server.example.com
...
```

  IdM サーバーがオフラインになると、IdM クライアントの SSSD サービスが、自動的に検出した別の IdM サーバーに接続します。

- パフォーマンス上の理由から DNS ルックアップをバイパスする場合は、 **ipa_server** パラメーターから **_srv_** エントリーを削除し、クライアントが接続すべき IdM サーバーを優先順に指定します。

```
[root@client ~]# cat /etc/sssd/sssd.conf

[domain/example.com]
id_provider = ipa
ipa_server = server1.example.com, server2.example.com
...
```

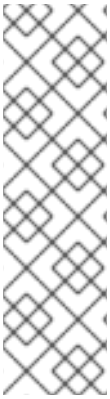
4.2. サーバー側の負荷分散およびサービスの可用性

複数の IdM レプリカをインストールして、IdM で負荷分散および高可用性を実行できます。

- 地理的に分散したネットワークがある場合には、データセンターごとに複数の IdM レプリカを設定することで、IdM クライアントと、最寄りのアクセス可能なサーバーとの間のパスを短くできます。

- Red Hat は、最大 60 台のレプリカを使用する環境をサポートします。

- IdM レプリケーションメカニズムでは、アクティブ/アクティブのサービスの可用性 (全 IdM レプリカのサービスを同時利用可) を提供します。



注記

Red Hat は、IdM およびその他の負荷分散または高可用性 (HA) ソフトウェアを組み合わせることを推奨します。

サードパーティーの高可用性ソリューションの多くは、アクティブ/パッシブのシナリオを想定しており、IdM へのサービスが不要に中断されてしまう可能性があります。他のソリューションでは、クラスター化されたサービスごとに仮想 IP または単一のホスト名を使用します。このような方法はすべて、通常、IdM ソリューションが提供するタイプのサービスの可用性では適切に機能しません。また、Kerberos との統合性が非常に低く、デプロイメントのセキュリティーと安定性が全体的に低下します。

第5章 レプリカトポロジーの最適化

堅牢なレプリカトポロジーはワークロードを分散し、レプリケーションの遅延を減らします。以下のガイドラインに従って、レプリカトポロジーのレイアウトを最適化します。

5.1. トポロジー内の IDM レプリカの適切な数を決定するためのガイドライン

組織の要件に合わせて IdM トポロジーを計画し、最適なパフォーマンスとサービスの可用性を確保してください。

各データセンターに少なくとも 2 つのレプリカをセットアップする

各データセンターに少なくとも 2 つのレプリカをデプロイして、1 台のサーバーに障害が発生した場合にレプリカが引き継いで要求を処理できるようにします。

クライアントにサービスを提供するために十分な数のサーバーを設定

1 台の Identity Management (IdM) サーバーで 2000 ~ 3000 台のクライアントにサービスを提供できます。ここでは、クライアントがサーバーに対して 1 日に複数回クエリーする (毎分ではありません) ことを想定しています。より頻繁なクエリーが予想される場合は、より多くのサーバーを計画してください。

十分な数の認証局 (CA) レプリカを設定します。

CA ロールがインストールされているレプリカのみが、証明書データを複製できます。IdM CA を使用する場合は、環境に、証明書のレプリカ合意がある CA レプリカが 2 つ以上あることを確認します。

1 つの IdM ドメインに最大 60 台のレプリカを設定

Red Hat は、最大 60 のレプリカを持つ環境に対応します。

5.2. トポロジーで IDM レプリカを接続するためのガイドライン

1 台のレプリカを少なくとも 2 つのレプリカに接続

追加のレプリカ合意を設定すると、初期レプリカと最初にインストールしたサーバーとの間だけでなく、他のレプリカ間でも情報が複製されます。

レプリカを、その他のレプリカ (最大 4 つ) に接続 (必須要件ではありません)

サーバーごとに多数のレプリカ合意を設定しても、大きな利点はありません。受信側のレプリカは、一度に 1 つの他のレプリカによってのみ更新できます。その間、その他のレプリカ合意はアイドル状態になります。通常、レプリカごとに 4 つ以上のレプリカ合意があると、リソースが無駄になります。

注記

この推奨事項は、証明書のレプリケーションとドメインのレプリケーションの両方に適用されます。

レプリカごとに 4 つのレプリカ合意という制限は、次の 2 つの場合には、例外として適用されません。

- 特定のレプリカがオンラインでない場合や応答していない場合にフェイルオーバーパスが必要な場合
- 大規模デプロイメントで、特定のノード間に追加の直接リンクが必要な場合

レプリカ合意を多数設定すると、全体のパフォーマンスに悪影響が及ぶ可能性があります。トポロジー内の複数のレプリカ合意が更新を送信すると、特定のレプリカの changelog データベースファイル上で、受信する更新と送信する更新の間の競合が増大することがあります。

レプリカごとにさらに多くのレプリケーションアグリーメントを使用する場合は、レプリケーションの問題やレイテンシーが発生しないようにしてください。距離が長く、中間ノードの数が多いと、レイテンシーの問題が発生する場合があることに注意してください。

データセンター内のレプリカを互いに接続

これにより、データセンター内のドメインレプリケーションが確実になります。

各データセンターを少なくとも 2 つの他のデータセンターに接続

これにより、データセンター間のドメインレプリケーションが確実になります。

少なくとも一対のレプリカ合意を使用してデータセンターを接続

データセンター A および B に、A1 への B1 までのレプリカ合意がある場合は、A2 から B2 へのレプリカ合意があれば、いずれかのサーバーがダウンしても、2 つのデータセンター間でレプリケーションを続行できます。

5.3. レプリカトポロジーの例

次のいずれかの例を使用して、信頼性の高いレプリカトポロジーを作成できます。

図5.1 4 つのデータセンターで構成されるレプリカトポロジー。各データセンターに、レプリカ合意で接続された 4 台のサーバーがある

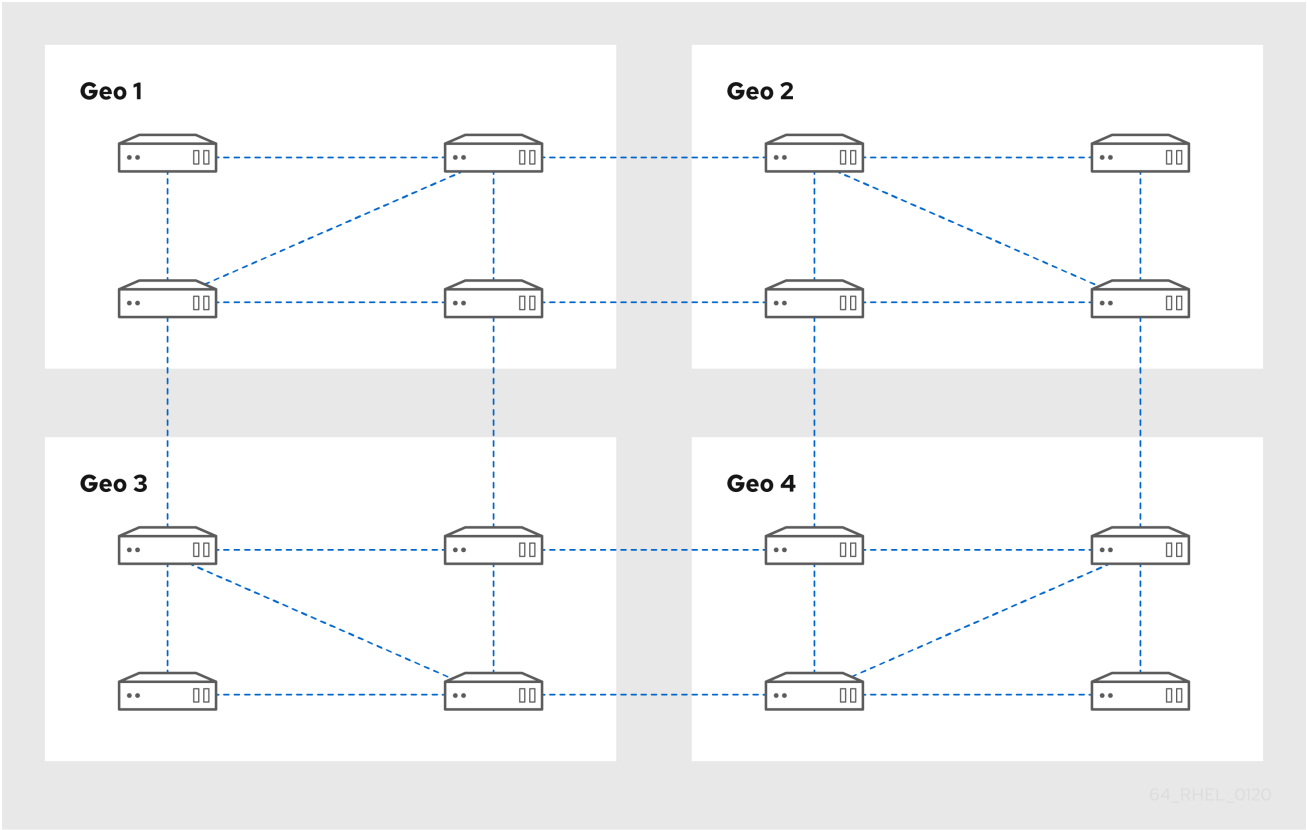
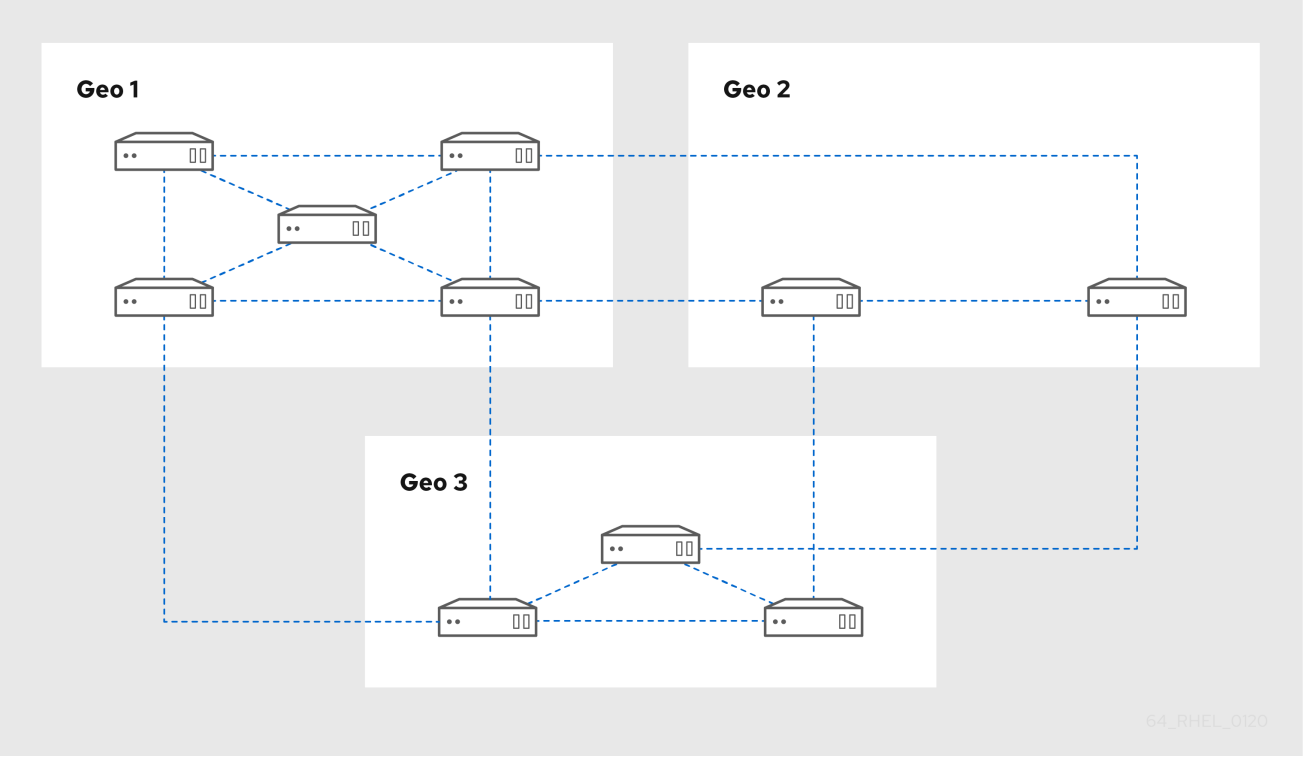
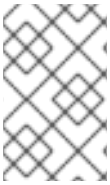


図5.2 3 つのデータセンターで構成されるレプリカトポロジー。各データセンターに異なる数のサーバーがあり、それらがすべてレプリカ合意を通じて相互接続されている



5.4. IDM サーバーからの IDM CA サービスのアンインストール

トポロジー内に **CA ロール** を持つ Identity Management (IdM) レプリカが 5 つ以上あり、冗長な証明書のレプリケーションが原因でパフォーマンスの問題が発生する場合、(RH) は IdM レプリカから冗長な CA サービスインスタンスを削除することを推奨します。そのためには、当該 IdM レプリカの使用を完全に停止してから、CA サービスを使用せずに IdM を再インストールする必要があります。



注記

IdM レプリカに CA ロールを **追加** することはできますが、IdM では、IdM レプリカから CA ロールのみを **削除** する方法はありません。**ipa-ca-install** コマンドには **--uninstall** オプションがありません。

前提条件

- トポロジー内の 5 つ以上の IdM サーバーに IdM CA サービスがインストールされている。

手順

1. 冗長な CA サービスを特定し、当該サービスをホストする IdM レプリカで IdM サーバーのアンインストール の手順を実行します。
2. 同じホストで、IdM サーバーのインストール: 統合 DNS があり外部 CA がない場合 の手順に従います。

5.5. 関連情報

- レプリカトポロジーの計画
- レプリケーショントポロジーの管理

第6章 検索サイズおよび時間制限の調整

IdM ユーザーのリストを要求するなど、一部のクエリーでは、エントリー数が大量に返される場合があります。この検索操作を調整して、**ipa user-find** などの **ipa *-find** コマンドの実行時や、Web UI で対応するリストを表示する際に、全体的なサーバーのパフォーマンスを向上できます。

検索サイズ制限

クライアントの CLI または IdM Web UI にアクセスするブラウザーからサーバーに送信されるリクエストで返される最大エントリー数を定義します。
デフォルト - 100 エントリー

検索時間の制限

検索の実行までにサーバーが待機する最大時間 (秒) を定義します。検索がこの制限に到達したら、サーバーは検索を停止し、停止するまでの期間に検出されたエントリーを返します。
デフォルト - 2 秒

この値が **-1** に設定されていると、IdM は、検索時に制限を適用しません。

重要

検索のサイズや時間制限を高く設定しすぎると、サーバーのパフォーマンスに影響を及ぼすことがあります。

6.1. コマンドラインで検索サイズおよび時間制限の調整

以下の手順では、コマンドラインで検索サイズと時間制限を調整する方法について説明します。

- グローバル
- 特定のエントリーの場合

手順

1. 現在の検索時間およびサイズ制限を CLI で表示するには、**ipa config-show** コマンドを使用します。

```
$ ipa config-show

Search time limit: 2
Search size limit: 100
```

2. すべてのクエリーに対して **グローバルに** 制限を調整するには、**ipa config-mod** コマンドを使用して、**--searchrecordslimit** および **--searchtimelimit** のオプションを追加します。以下に例を示します。

```
$ ipa config-mod --searchrecordslimit=500 --searchtimelimit=5
```

3. 特定のクエリーに対してのみ **一時的に** 制限を調整するには、コマンドに **--sizelimit** または **--timelimit** オプションを追加してください。以下に例を示します。

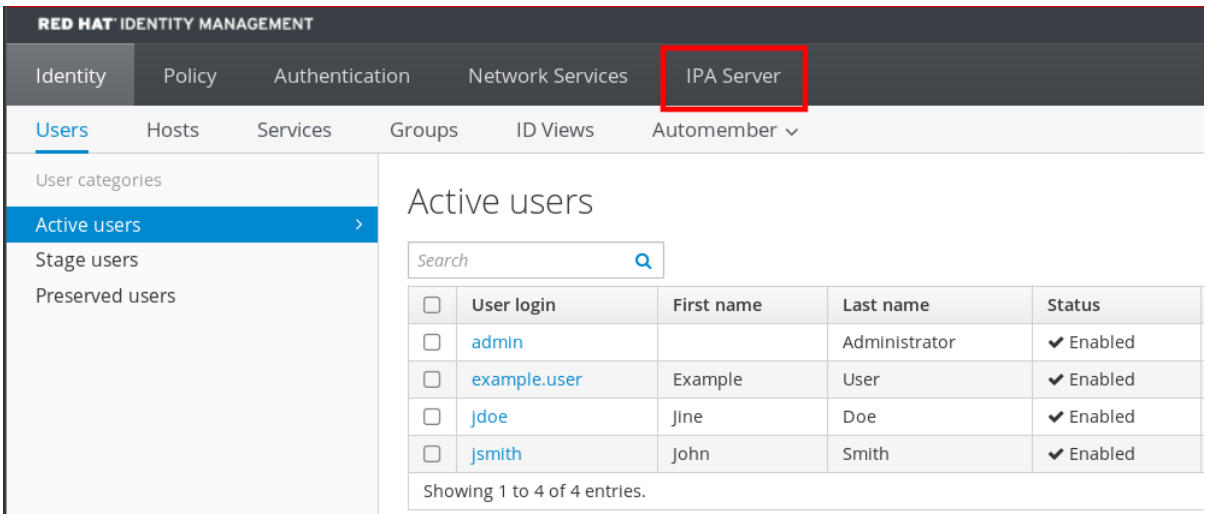
```
$ ipa user-find --sizelimit=200 --timelimit=120
```

6.2. WEB UI で検索サイズおよび時間制限の調整

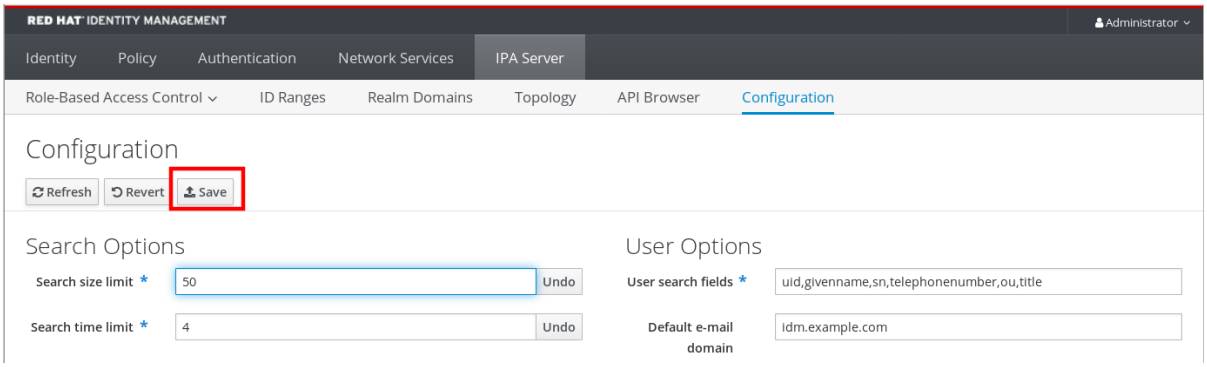
以下の手順では、IdM Web UI でグローバル検索のサイズと時間制限を調整する方法について説明します。

手順

1. IdM Web UI にログインします。
2. **IPA Server** をクリックします。



3. **IPA Server** タブで、**Configuration** をクリックします。
4. **Search Options** エリアに必要な値を設定します。
   デフォルト値は以下の通りです。
   - 検索サイズの制限 - 100 エントリー
   - 検索時間の制限 - 2 秒
5. ページ上部にある **Save** をクリックします。



第7章 ADJUSTING IDM DIRECTORY SERVER PERFORMANCE

Directory Server のリソースと動作を制御する LDAP 属性を調整して、Identity Management のデータベースのパフォーマンスをチューニングできます。

Directory Server による **データのキャッシュ** 方法を調整するには、以下の手順を参照してください。

- エントリーキャッシュサイズの調整
- データベースインデックスキャッシュサイズの調整
- エントリーおよびデータベースキャッシュサイズの自動調整の再有効化
- DN キャッシュサイズの調整
- 正規化された DN キャッシュサイズの調整

Directory Server の **リソース制限** を調整するには、以下の手順を参照してください。

- メッセージの最大サイズの調整
- ファイル記述子の最大数の調整
- 接続バックログサイズの調整
- データベースロックの最大数の調整

パフォーマンスに最も影響を与える **タイムアウト** を調整するには、以下の手順を参照してください。

- 入出力ブロックのタイムアウトの調整
- アイドル接続タイムアウトの調整
- レプリケーションリリースタイムアウトの調整

LDIF ファイルからカスタム Directory Server 設定を使用して IdM サーバーまたはレプリカをインストールするには、次の手順を参照してください。

- LDIF ファイルからのカスタムデータベース設定を使用した IdM サーバーまたはレプリカのインストール

7.1. エントリーキャッシュサイズの調整

重要

Red Hat は、パフォーマンスの最適化にビルドインのキャッシュサイズの自動調整機能を使用することを推奨します。auto-tuned の値を意図的に変更する必要がある場合を除き、この値は変更しないでください。

nsslapd-cachememsize 属性は、エントリーキャッシュで利用可能なメモリー領域のサイズ (バイト単位) を指定します。この属性は、ディレクトリーサーバーが使用する物理メモリーのサイズ制御で最も重要な値です。

エントリーキャッシュサイズが小さすぎると、**/var/log/dirsrv/slapd-INSTANCE-NAME/errors** ログファイルの Directory Server エラーログに以下のエラーが表示される場合があります。

-

> REASON: entry too large (**83886080** bytes) for the import buffer size (**67108864** bytes). **Try increasing nsslapd-cachememsize.**

Red Hat は、メモリー内でエントリーキャッシュとデータベースインデックスエントリーキャッシュを適合させることを推奨します。

デフォルト値	**209715200** (200 MiB)
有効な範囲	**500000 - 18446744073709551615** (500 kB - (2^{64}-1))
エントリー DN の場所	**cn=database-name,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. 自動キャッシュチューニングを無効にします。

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

2. データベースの接尾辞と、対応するバックエンドを表示します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix list
cn=changelog (changelog)
dc=example,dc=com (userroot)
o=ipaca (ipaca)
```

このコマンドにより、各接尾辞の横にバックエンドデータベースが表示されます。次の手順では、接尾辞のデータベース名を使用します。

3. データベースのエントリーキャッシュサイズを設定します。この例では、userroot データベースのエントリーキャッシュを 2 ギガバイトに設定します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix set --cache-memsize=2147483648 userroot
```

4. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

5. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **cache-memsize** を別の値に調整するか、キャッシュサイズの自動調整をもう一度有効化します。

検証

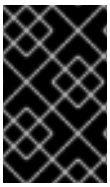
- **nsslapd-cachememsize** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-
cachememsize
nsslapd-cachememsize: 2147483648
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-cachememsize
- エントリーおよびデータベースキャッシュの自動サイズ設定の再有効化

7.2. データベースインデックスキャッシュサイズの調整



重要

Red Hat は、パフォーマンスの最適化にビルドインのキャッシュサイズの自動調整機能を使用することを推奨します。auto-tuned の値を意図的に変更する必要がある場合を除き、この値は変更しないでください。

nsslapd-dbcachesize 属性は、データベースインデックスが使用するメモリー量を制御します。このキャッシュサイズは、エントリーキャッシュサイズと比べ、Directory Server パフォーマンスへの影響は少なくなっていますが、エントリーキャッシュサイズの設定後に利用可能なメモリーがある場合には、Red Hat は、データベースキャッシュに割り当てるメモリー量を増やすことを推奨します。

データベースキャッシュのメモリーの上限は 1.5 GB で、これ以上の値を指定してもパフォーマンスが改善されないためです。

デフォルト値	**10000000** (10 MB)
有効な範囲	**500000 - 1610611911** (500 kB - 1.5GB)
エントリー DN の場所	**cn=config,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. 自動キャッシュチューニングを無効にし、データベースキャッシュのサイズを設定します。この例では、データベースキャッシュを 256 メガバイトに設定します。

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

2. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

3. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **dbcachesize** を別の値に調整するか、キャッシュサイズの自動調整をもう一度有効化します。

検証

- **nsslapd-dbcachesize** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-dbcachesize
nsslapd-dbcachesize: 2147483648
```

関連情報

- Directory Server 11 ドキュメント の nsslapd-dbcachesize
- エントリーおよびデータベースキャッシュの自動サイズ設定の再有効化

7.3. データベースおよびエントリーキャッシュサイズの自動調整の再有効化

重要

Red Hat は、パフォーマンスの最適化にビルドインのキャッシュサイズの自動調整機能を使用することを推奨します。Red Hat では、キャッシュサイズの手動設定は推奨していません。

デフォルトでは、IdM Directory Server は、データベースキャッシュとエントリーキャッシュに最適なサイズを自動的に判断します。自動サイズ調整は、空きメモリーの一部だけを残し、インスタンスの起動時にサーバーのハードウェアリソースに基づいて、出たベースおよびエントリー両方のキャッシュのサイズを最適化します。

以下の手順を使用して、カスタムのデータベースキャッシュおよびエントリーキャッシュの値を元に戻し、キャッシュの自動サイズ調整機能をデフォルト値に戻します。

nsslapd-cache-autosize	この設定では、データベースおよびエントリーキャッシュの自動サイズ調整に割り当てる空きメモリー容量を制御します。値が **0** の場合は、自動サイズ調整が無効になります。
デフォルト値	**10** (空きメモリーの 10%)
有効な範囲	**0 - 100**
エントリー DN の場所	**cn=config,cn=ldbm database,cn=plugins,cn=config**

nsslapd-cache-autosize-split	この値は、**nsslapd-cache-autosize** で決定する空きメモリーの割合を設定します。この値は、データベースキャッシュに使用されます。残りのメモリーはエントリーキャッシュに使用されます。

デフォルト値	**25** (データベースキャッシュに 25%、エントリーキャッシュに 60%)
有効な範囲	**0 - 100**
エントリー DN の場所	**cn=config,cn=ldbm database,cn=plugins,cn=config**

前提条件

- 以前にデータベースおよびエントリーキャッシュの自動調整を無効にしている。

手順

1. Directory Server を停止します。

```
[root@server ~]# systemctl stop dirsrv.target
```

2. **/etc/dirsrv/slapd-instance_name/dse.ldif** ファイルをバックアップしてから、他の変更を加えます。

```
[root@server ~]# *cp /etc/dirsrv/slapd-instance_name/dse.ldif \
    /etc/dirsrv/slapd-instance_name/dse.ldif.bak.$(date "+%F_%H-%M-%S")
```

3. **/etc/dirsrv/slapd-instance_name/dse.ldif** ファイルを編集します。

   a. データベースおよびエントリーキャッシュに使用する空きシステムメモリーの割合を、デフォルトの空きメモリー 10% に戻します。

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

   b. データベースキャッシュのシステムメモリーから使用する割合をデフォルトの 25% に設定します。

```
nsslapd-cache-autosize-split: 25
```

4. **/etc/dirsrv/slapd-instance_name/dse.ldif** ファイルへの変更を保存します。

5. Directory Server を起動します。

```
[root@server ~]# systemctl start dirsrv.target
```

検証

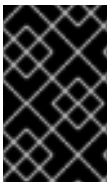
- **nsslapd-cache-autosize** および **nsslapd-cache-autosize-split** 属性の値を表示して、任意の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-cache-autosize
nsslapd-cache-autosize: *10
nsslapd-cache-autosize-split: 25
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-cache-autosize

7.4. DN キャッシュサイズの調整



重要

Red Hat は、パフォーマンスの最適化にビルドインのキャッシュサイズの自動調整機能を使用することを推奨します。auto-tuned の値を意図的に変更する必要がある場合を除き、この値は変更しないでください。

nsslapd-dncachememsize 属性は、識別名 (DN) キャッシュで利用可能なメモリー領域のサイズ (バイト単位) を指定します。DN キャッシュはデータベースのエントリーキャッシュと似ていますが、テーブルにはエントリー ID とエントリー DN のみが保存され、**rename** および **moddn** 操作のルックアップ時間を短縮できます。

デフォルト値	**10485760** (10 MB)
有効な範囲	**500000 - 18446744073709551615** (500 kB - (2^{64}-1))
エントリー DN の場所	**cn=database-name,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **(オプション)** データベースの接尾辞と、対応する d データベース名を表示します。

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

このコマンドにより、各接尾辞の横にバックエンドデータベースが表示されます。次の手順では、接尾辞のデータベース名を使用します。

2. データベースの DN キャッシュサイズを設定します。この例では、DN キャッシュを 20 メガバイトに設定します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend suffix set --dncache-memsize=20971520 userroot
```

3. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **dncache-memsize** を別の値に調整するか、デフォルトの 10 MB に戻します。

検証

- **nsslapd-dncachememsize** 属性の新しい値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=userroot,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-
dncachememsize
nsslapd-dncachememsize: 20971520
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-dncachememsize

7.5. 正規化された DN キャッシュサイズの調整

重要

Red Hat は、パフォーマンスの最適化にビルドインのキャッシュサイズの自動調整機能を使用することを推奨します。auto-tuned の値を意図的に変更する必要がある場合を除き、この値は変更しないでください。

nsslapd-ndn-cache-max-size 属性は、通常の識別名 (NDN) を格納するキャッシュサイズ (バイト単位) を制御します。この値を増やすと、メモリーに頻繁に使用される DN を確保します。

デフォルト値	**20971520** (20 MB)
有効な範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. NDN キャッシュが有効になっていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-ndn-cache-enabled
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-enabled: on
```

キャッシュが **オフ** の場合は、次のコマンドで有効にします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
```

```
replace nsslapd-ndn-cache-enabled=on
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-ndn-cache-enabled"
```

2. **nsslapd-ndn-cache-max-size** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。プロンプトが表示されたら、Directory Manager のパスワードを入力します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-ndn-cache-max-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-max-size: 20971520
```

3. **nsslapd-ndn-cache-max-size** 属性の値を変更します。この例では **41943040** (40 MB) に値を増やします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-ndn-cache-max-size=41943040
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **nsslapd-ndn-cache-max-size** を別の値に調整するか、キャッシュサイズの自動調整をもう一度有効化します。

検証

- **nsslapd-ndn-cache-max-size** 属性の新しい値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-ndn-cache-max-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ndn-cache-max-size: 41943040
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-ndn-cache-max-size

7.6. メッセージの最大サイズの調整

nsslapd-maxbersize 属性は、受信メッセージまたは LDAP 要求での最大許容サイズをバイト単位で設定します。要求のサイズを制限すると、さまざまな DoS 攻撃を防ぎます。

最大メッセージサイズが小さすぎると、**/var/log/dirsrv/slapd-INSTANCE-NAME/errors** の Directory Server エラーログに以下のエラーが表示される可能性があります。

```
Incoming BER Element was too long, max allowable is 2097152 bytes. Change the nsslapd-maxbersize attribute in cn=config to increase.
```

この制限の対象は LDAP 要求の合計サイズです。たとえば、エントリーの追加要求で、要求のエントリーがデフォルトの設定値よりも大きい場合に、この追加要求は拒否されます。ただし、この制限はレプリケーションプロセスには適用されません。この属性の変更には細心の注意を払ってください。

デフォルト値	**2097152** (2 MB)
有効な範囲	**0 - 2147483647** (0 から 2 GB)
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-maxbersize** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。プロンプトが表示されたら、Directory Manager のパスワードを入力します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 2097152
```

2. **nsslapd-maxbersize** 属性の値を変更します。この例では **4194304** (4 MB) に値を増やします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-maxbersize=4194304
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-maxbersize"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **nsslapd-maxbersize** を別の値に調整するか、デフォルトの **2097152** に戻します。

検証

- **nsslapd-maxbersize** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-maxbersize
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxbersize: 4194304
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-maxbersize (最大メッセージサイズ)

7.7. ファイル記述子の最大数の調整

/etc/systemd/system.conf ファイルの **DefaultLimitNOFILE** パラメーターに値を定義できます。**root** 権限を持つ管理者は、**setrlimit** コマンドを使用して、**ns-slapd** プロセスの **DefaultLimitNOFILE** パラメーターをより低い値に設定できます。この値は **/etc/systemd/system.conf** の値よりも優先され、Identity Management (IdM) Directory Server (DS) によって **nsslapd-maxdescriptors** 属性の値として受け入れられます。

nsslapd-maxdescriptors 属性は、IdM LDAP が使用するファイル記述子の最大数 (プラットフォームに依存) を設定します。ファイル記述子は、クライアント接続、ログファイル、ソケットなどのリソースに使用されます。

/etc/systemd/system.conf または **setrlimit** のいずれにも値が定義されていない場合、IdM DS は **nsslapd-maxdescriptors** 属性を 1048576 に設定します。

IdM DS 管理者が後で **nsslapd-maxdescriptors** に新しい値を手動で設定した場合、IdM DS は新しい値を、**setrlimit** または **/etc/systemd/system.conf** でローカルに定義された値と比較します。その結果に応じて、以下を実行します。

- **nsslapd-maxdescriptors** の新しい値がローカルで定義されている値よりも大きい場合、サーバーは新しい値の設定を拒否し、ローカルの制限値を高基準値として引き続き適用します。
- 新しい値がローカルで定義されている値よりも小さい場合、新しい値が使用されます。

この手順では、**nsslapd-maxdescriptors** に新しい値を設定する方法について説明します。

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-maxdescriptors** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。プロンプトが表示されたら、Directory Manager のパスワードを入力します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-maxdescriptors
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 4096
```

2. **nsslapd-maxdescriptors** 属性の値を変更します。この例では、値を **8192** に増やします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
replace nsslapd-maxdescriptors=8192
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-maxdescriptors"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **nsslapd-maxdescriptors** を別の値に調整するか、デフォルトの **4096** に戻します。

検証

- **nsslapd-maxdescriptors** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-maxdescriptors
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-maxdescriptors: 8192
```

関連情報

- Directory Server 12 ドキュメントの nsslapd-maxdescriptors (最大ファイル記述子)

7.8. 接続バックログサイズの調整

listen サービスは、受信接続を受け付けるソケット数を設定します。**nsslapd-listen-backlog-size** の値は、接続を拒否する前の **sockfd** ソケットのキューの最大長を設定します。

IdM 環境で大量の接続を処理する場合は、**nsslapd-listen-backlog-size** の値を増やすことを検討してください。

デフォルト値	キュースロット **128**
有効な範囲	**0 - 9223372036854775807**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-listen-backlog-size** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。プロンプトが表示されたら、Directory Manager のパスワードを入力します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-listen-backlog-size: 128
```

2. **nsslapd-listen-backlog-size** 属性の値を変更します。この例では、値を **192** に増やします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
replace nsslapd-listen-backlog-size=192
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-listen-backlog-size"
```

検証

- **nsslapd-listen-backlog-size** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-listen-backlog-size
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-listen-backlog-size: 192
```

関連情報

- Directory Server 11 のドキュメントの nsslapd-listen-backlog-size

7.9. データベースロックの最大数の調整

ロックメカニズムでは、同時に実行できる Directory Server プロセスのコピー数を制御し、**nsslapd-db-locks** パラメーターは最大ロック数を設定します。

/var/log/dirsrv/slapd-instance_name/errors ログファイルに以下のエラーメッセージが表示される場合に、最大ロック数を増やします。

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

デフォルト値	ロック数 **50000**
有効な範囲	**0 - 2147483647**
エントリー DN の場所	**cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-db-locks** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword -b "cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-db-locks
nsslapd-db-locks: 50000
```

2. **locks** 属性の値を変更します。この例では、ロックの値を 2 倍の **100000** に設定します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --locks=100000
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully updated database configuration
```

4. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

検証

- **nsslapd-db-locks** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
-b "cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config" | grep nsslapd-db-
locks
nsslapd-db-locks: 100000
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-db-locks

7.10. 入出力ブロックのタイムアウトの調整

nsslapd-ioblocktimeout 属性は、停止した LDAP クライアントへの接続が切断されるまでの時間をミリ秒単位で設定します。LDAP クライアントは、読み取りまたは書き込み操作の I/O の進捗が全くない場合には停止されたと見なされます。

nsslapd-ioblocktimeout 属性の値を減らして、できるだけ早い段階で接続のロックを解除します。

デフォルト値	**10000** ミリ秒
有効な範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-ioblocktimeout** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。プロンプトが表示されたら、Directory Manager のパスワードを入力します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-ioblocktimeout: 10000
```

2. **nsslapd-ioblocktimeout** 属性の値を変更します。この例では、値が **8000** まで減らします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace nsslapd-ioblocktimeout=8000
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-ioblocktimeout"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **nsslapd-ioblocktimeout** を別の値に調整するか、デフォルトの **10000** に戻します。

検証

- **nsslapd-ioblocktimeout** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config get nsslapd-ioblocktimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 8000
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-ioblocktimeout (IO ブロックタイムアウト)

7.11. アイドル接続タイムアウトの調整

nsslapd-idletimeout 属性は、アイドル状態の LDAP クライアントの接続が IdM サーバーにより切断されるまでの時間を秒単位で設定します。値が **0** の場合は、サーバーはアイドル状態の接続を切断しません。

Red Hat は、この値を調節して使用されなくなった接続を切断することを推奨しますが、有効な接続が早い段階で切断されることはありません。

デフォルト値	**3600** 秒 **(1 時間)**
有効な範囲	**0 - 2147483647**
エントリー DN の場所	**cn=config**

前提条件

- LDAP Directory Manager のパスワード

手順

1. **nsslapd-idletimeout** パラメーターの現在の値を取得して、復元する必要がある場合に備え、調整を行う前にこの値をメモします。プロンプトが表示されたら、Directory Manager のパスワードを入力します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
```

```
get nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

2. **nsslapd-idletimeout** 属性の値を変更します。この例では、値が **1800** (30 分) まで減らします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
replace nsslapd-idletimeout=1800
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "nsslapd-idletimeout"
```

4. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **nsslapd-idletimeout** を別の値に調整するか、デフォルトの **3600** に戻します。

検証

- **nsslapd-idletimeout** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com config
get nsslapd-idletimeout
Enter password for cn=Directory Manager on ldap://server.example.com:
nsslapd-idletimeout: 3600
```

関連情報

- Directory Server 11 ドキュメントの nsslapd-idletimeout (デフォルトのアイドルタイムアウト)

7.12. レプリケーションリリースタイムアウトの調整

IdM レプリカは、別のレプリカでのレプリケーションセッション中は同時に使用できないのでロックされます。一部の環境では、大規模な更新やネットワークの輻輳により、レプリカが長時間ロックされるため、レプリケーションのレイテンシーが増加します。

repl-release-timeout パラメーターを調整すると、一定時間が経過すると、レプリカをリリースできます。Red Hat は、この値を **30** から **120** の間に設定することを推奨します。

- 値を低く設定しすぎると、レプリカ間でお互いを取得しあうため、大規模な更新を送信できなくなります。
- タイムアウトが長くなると、サーバーが長時間 1 つのレプリカだけにアクセスするのが最適な場合など、トラフィックが多い状況が改善されますが、**120** 秒よりも長く設定すると、レプリケーションに時間がかかります。

デフォルト値	**60** 秒
有効な範囲	**0 - 2147483647**

推奨される範囲	**30 - 120**

前提条件

- LDAP Directory Manager のパスワード

手順

1. データベースの接尾辞と、対応するバックエンドを表示します。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com
backend suffix list
cn=changelog (changelog)
dc=example,dc=com (userroot)
o=ipaca (ipaca)
```

   このコマンドにより、接尾辞の横にバックエンドデータベースの名前が表示されます。次の手順でこの接尾辞名を使用します。

2. メインの userroot データベースの **repl-release-timeout** 属性の値を変更します。この例では、値を **90** 秒に増やします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com
replication set --suffix="dc=example,dc=com" --repl-release-timeout=90
```

3. Directory Manager として認証し、設定の変更を行います。

```
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "repl-release-timeout"
```

4. (オプション) IdM 環境で IdM 認証局 (CA) を使用する場合は、CA データベースの **repl-release-timeout** 属性の値を変更できます。この例では、値を **90** 秒に増やします。

```
[root@server ~]# dsconf -D "cn=Directory Manager" ldap://server.example.com replication
set --suffix="o=ipaca" --repl-release-timeout=90
Enter password for cn=Directory Manager on ldap://server.example.com:
Successfully replaced "repl-release-timeout"
```

5. Directory Server を再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

6. IdM ディレクトリーサーバーのパフォーマンスを監視します。希望どおりに変更されなかった場合にはこの手順を繰り返して **repl-release-timeout** を別の値に調整するか、デフォルトの **60** に戻します。

検証

- **nsds5ReplicaReleaseTimeout** 属性の値を表示し、希望の値に設定されていることを確認します。

```
[root@server ~]# ldapsearch -D "cn=directory manager" -w DirectoryManagerPassword
```

```
-b "cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=mapping tree,cn=config" | grep
nsds5ReplicaReleaseTimeout
nsds5ReplicaReleaseTimeout: 90
```

注記

この例では、接尾辞の識別名 (DN) は **dc=example,dc=com** ですが、**ldapsearch** コマンドでは等号記号 (**=**) とコンマ (**,**) をエスケープする必要があります。

接尾辞 DN を **cn=dc\3Dexample\2Cdc\3Dcom** に変換します。ここでは、以下のエスケープ文字を使用します。

- **=** は **\3D** に、
- **,** は **\2C** に置き換えます。

関連情報

- Directory Server 11 ドキュメントの nsDS5ReplicaReleaseTimeout

7.13. LDIF ファイルからのカスタムデータベース設定を使用した IDM サーバーまたはレプリカのインストール

Directory Server データベースのカスタム設定を使用して、IdM サーバーおよび IdM レプリカをインストールできます。以下の手順は、データベース設定で LDAP データ交換形式 (LDIF) ファイルを作成する方法と、その設定を IdM サーバーおよびレプリカインストールコマンドに渡す方法を示しています。

前提条件

- IdM 環境のパフォーマンスを向上させるカスタムの Directory Server 設定を行っている。IdM Directory Server パフォーマンスの調整 を参照してください。

手順

1. カスタムデータベース設定で LDIF 形式のテキストファイルを作成します。LDAP 属性の変更はダッシュ (-) で区切ります。この例では、idle タイムアウトおよび最大ファイルディスクリプターにデフォルト以外の値を設定します。

```
dn: cn=config
changetype: modify
replace: nsslapd-idletimeout
nsslapd-idletimeout=1800
-
replace: nsslapd-maxdescriptors
nsslapd-maxdescriptors=8192
```

2. **--dirsrv-config-file** パラメーターを使用して、LDIF ファイルをインストールスクリプトに渡します。

   a. IdM サーバーをインストールするには、次のコマンドを実行します。

```
# ipa-server-install --dirsrv-config-file filename.ldif
```

b. IdM レプリカをインストールするには、次のコマンドを実行します。

```
# ipa-replica-install --dirsrv-config-file filename.ldif
```

関連情報

- **ipa-server-install** コマンドおよび **ipa-replica-install** コマンドのオプション

7.14. 関連情報

- Directory Server 11 パフォーマンスチューニングガイド

第8章 KDC のパフォーマンスの調整

以下のセクションでは、ユーザー、ホスト、およびサービスの認証を行う Kerberos Key Distribution Center (KDC) のパフォーマンスを調整する方法を説明します。

8.1. KDC リッスンキューの長さの調整

/var/kerberos/krb5kdc/kdc.conf ファイルの **[kdcdefaults]** セクションで **kdc_tcp_listen_backlog** オプションを設定することにより、KDC デーモンのリッスンキューの長さのサイズを調整できます。**5** のデフォルト値は、大量の Kerberos トラフィックが発生する IdM デプロイメントでは低すぎる可能性がありますが、この値を高く設定しすぎるとパフォーマンスが低下します。

デフォルト値	**5**
有効な範囲	**1 - 10**

手順

1. テキストエディターで **/var/kerberos/krb5kdc/kdc.conf** ファイルを開きます。
2. TCP リッスンバックログを **7** などの目的の値に設定します。

```
[kdcdefaults]
 ...
 kdc_tcp_listen_backlog = 7
```

3. **/var/kerberos/krb5kdc/kdc.conf** ファイルを保存して閉じます。
4. KDC を再起動して、新しい設定を読み込みます。

8.2. レルムごとの KDC の動作制御オプション

KDC は認証の成功および失敗後にデータベースに書き込み、各 Kerberos レルムのユーザーアカウントのロックとロック解除を追跡します。**/etc/krb5.conf** ファイルの **[dbmodules]** セクションで以下のオプションを調整すると、KDC の情報の書き込みの頻度を最小限して、パフォーマンスを向上させることができます。

disable_last_success

true に設定すると、事前認証が必要なプリンシパルエントリーの **最後の正常な認証** フィールドまで KDC の更新が行われないようにします。

デフォルト値	**false**
有効な範囲	**true** または **false**

disable_lockout

true に設定すると、事前認証が必要なプリンシパルエントリーの **最後に失敗した認証** と **パスワードの失敗試行回数** フィールドまで KDC の更新が行われないようにします。このフラグを設定するとパフォーマンスが向上しますが、アカウントロックアウトを無効にすると、セキュリティーリスクと見なされる可能性があります。

デフォルト値	**false**
有効な範囲	**true** または **false**

関連情報

- レルムごとの KDC 設定の調整

8.3. レルムごとの KDC 設定の調整

この手順では、Kerberos レルムごとに KDC の動作を調整します。

手順

1. テキストエディターで **/etc/krb5.conf** ファイルを開きます。
2. **[dbmodules]** セクションとそれぞれの Kerberos レルムに、オプションと任意の値を指定します。この例では、Kerberos レルム **EXAMPLE.COM** に **disable_last_success** 変数を設定します。

```
[dbmodules]
   EXAMPLE.COM = {
      disable_last_success = true
   }
```

3. **/etc/krb5.conf** ファイルを保存して閉じます。
4. KDC を再起動して、新しい設定を読み込みます。

関連情報

- レルムごとの KDC の動作制御オプション

8.4. KRB5KDC プロセス数の調整

Key Distribution Center (KDC) が着信接続の処理を開始するプロセスの数を手動で調整するには、次の手順に従います。

デフォルトでは、IdM インストーラーは CPU コアの数を検出し、その値を **/etc/sysconfig/krb5kdc** ファイルに入力します。たとえば、ファイルには次のエントリーが含まれている場合があります。

```
KRB5KDC_ARGS='-w 2'
[...]
```

この例では、**KRB5KDC_ARGS** パラメーターを **-w 2** に設定すると、KDC は 2 つの別個のプロセスを開始して、メインプロセスからの着信接続を処理します。特に、要件に基づいて仮想 CPU の数を簡単に追加または削除できる仮想環境では、この値を調整することが推奨されます。ポート 88 の TCP/IP キューが増え続けてパフォーマンスの問題が発生したり、IdM サーバーが応答しなくなったりするのを防ぐには、**KRB5KDC_ARGS** パラメーターを手動で高い値に設定して、より多くのプロセスをシミュレートします。

手順

1. **/etc/sysconfig/krb5kdc** ファイルをテキストエディターで開きます。
2. **KRB5KDC_ARGS** パラメーターの値を指定します。この例では、プロセスの数を 10 に設定しています。

```
KRB5KDC_ARGS='-w 10'
[...]
```

3. **/etc/sysconfig/krb5kdc** ファイルを保存して閉じます。
4. systemd 設定をリロードします。

```
# systemctl daemon-reload
```

5. **krb5kdc** サービスを再起動します。

```
# systemctl restart krb5kdc.service
```

注記

IdM Healthcheck ユーティリティーを使用して、KDC が最適な数のワーカープロセスを使用するように設定されていることを確認できます。IdM Healthcheck を使用した KDC ワーカープロセスの最適な数の検証 を参照してください。

8.5. 関連情報

- MIT Kerberos ドキュメント - kdc.conf

第9章 大規模な IDM-AD 信頼デプロイメントのための SSSD パフォーマンスの調整

ユーザーおよびグループ情報の取得は、特に AD (System Security Services Daemon) ドメイン、つまり大規模な Active Directory (AD) ドメインへの信頼を持つ IdM デプロイメントでは、データ集中型の操作です。SSSD がアイデンティティプロバイダーから取得する情報とその期間を調整することで、このパフォーマンスを向上させることができます。

9.1. 大規模な IDM-AD 信頼デプロイメント向けの IDM サーバーでの SSSD の調整

この手順では、IdM サーバーの SSSD サービスの設定にチューニングオプションを適用し、大規模な AD 環境から情報を取得する時の応答時間が短縮します。

前提条件

- **/etc/sssd/sssd.conf** 設定ファイルを編集するには、**root** パーミッションが必要です。

手順

1. テキストエディターで **/etc/sssd/sssd.conf** 設定ファイルを開きます。
2. Active Directory ドメインの **[domain]** セクションに以下のオプションを追加します。AD ドメインにドメインセクションがない場合は作成します。

```
[domain/ad.example.com]
ignore_group_members = true
subdomain_inherit = ignore_group_members
...
```

3. サーバーの **/etc/sssd/sssd.conf** ファイルを保存して閉じます。
4. SSSD サービスを再起動して、設定の変更を読み込みます。

```
[root@client ~]# systemctl restart sssd
```

関連情報

- IdM-AD 信頼デプロイメント用の IdM サーバーおよびクライアントでの SSSD の調整オプション

9.2. IDM サーバーでの IPA-EXTDOM プラグインの設定タイムアウトの調整

IdM クライアントは Active Directory (AD) からユーザーとグループに関する情報を直接受信できないため、IdM サーバーは **ipa-extdom** プラグインを使用して AD ユーザーとグループに関する情報を受信し、その情報は要求元のクライアントに転送されます。

ipa-extdom プラグインは、AD ユーザーのデータに要求を SSSD に送信します。情報が SSSD キャッシュにない場合、SSSD は AD ドメインコントローラー (DC) にデータを要求します。config タイムアウト値を調整できます。これは、プラグインが接続をキャンセルして発信者にタイムアウトエラーを返す前に、**ipa-extdom** プラグインが SSSD からの応答を待機する時間を定義します。デフォルト値は 10000 ミリ秒 (10 秒) です。

次の例では、設定タイムアウトを 20 秒 (20000 ミリ秒) に調整します。

警告

設定タイムアウトを調整するときは注意してください。

- 設定する値が小さすぎる (例: 500 ミリ秒) と、SSSD に応答するのに十分な時間がない可能性があり、要求は常にタイムアウトを返します。
- 設定する値が大きすぎる (例: 30000 ミリ秒 (30 秒)) と、1 つの要求が、この期間、SSSD への接続をブロックする可能性があります。一度に SSSD に接続できるのは 1 つのスレッドであるため、プラグインからの他のリクエストはすべて待機する必要があります。
- IdM クライアントから送信された要求が多い場合、IdM サーバー上の Directory Server 用に設定された使用可能なすべてのワーカーをブロックできます。その結果、サーバーはしばらくの間、どのような種類の要求にも応答できない可能性があります。

以下の状況で設定のタイムアウトを変更します。

- AD ユーザーおよびグループに関する情報を要求する際に、独自の検索タイムアウトが発生する前に IdM クライアントが頻繁にタイムアウトエラーを受け取ると、設定のタイムアウト値が **小さすぎ** ます。
- IdM サーバーで Directory Server がロックされていることが多く、**pstack** ユーティリティーは、この時点で多数またはすべてのワーカースレッドが **ipa-extdom** 要求を処理していることを報告する場合は、値が **大きすぎ** ます。

前提条件

- LDAP Directory Manager のパスワード

手順

- 次のコマンドを使用して、設定タイムアウトを 20000 ミリ秒に調整します。

```
# ldapmodify -D "cn=directory manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxNssTimeout
ipaExtdomMaxNssTimeout: 20000
```

9.3. IDM サーバー上の IPA-EXTDOM プラグインの最大バッファーサイズの調整

IdM クライアントは Active Directory (AD) からユーザーとグループに関する情報を直接受信できないため、IdM サーバーは **ipa-extdom** プラグインを使用して AD ユーザーとグループに関する情報を受信し、その情報は要求元のクライアントに転送されます。

ipa-extdom プラグインの最大バッファーサイズを調整できます。これにより、SSSD が受信するデータを保存できるバッファーのサイズが調整されます。バッファーが小さすぎると、SSSD は **ERANGE** エラーを返し、プラグインはより大きなバッファーで要求を再試行します。デフォルトのバッファーサイズは 134217728 バイト (128 MB) です。

次の例では、最大バッファーサイズを 256 MB (268435456 バイト) に調整します。

前提条件

- LDAP Directory Manager のパスワード

手順

- 次のコマンドを使用して、最大バッファーサイズを 268435456 バイトに設定します。

```
# ldapmodify -D "cn=directory manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxNssBufSize
ipaExtdomMaxNssBufSize: 268435456
```

9.4. IDM サーバー上の IPA-EXTDOM プラグインの最大インスタンス数の調整

IdM クライアントは Active Directory (AD) からユーザーとグループに関する情報を直接受信できないため、IdM サーバーは **ipa-extdom** プラグインを使用して AD ユーザーとグループに関する情報を受信し、その情報は要求元のクライアントに転送されます。

デフォルトでは、**ipa-extdom** プラグインは LDAP ワーカースレッドの最大 80% を使用して IdM クライアントからのリクエストを処理するように設定されています。IdM クライアントの SSSD サービスが AD 信頼ユーザーおよびグループに関する大量の情報を要求した場合に、LDAP スレッドのほとんどを使用すると、この操作によって LDAP サービスが停止する可能性があります。これらの問題が発生した場合、AD ドメインの SSSD ログファイル **/var/log/sssd/sssd__your-ad-domain-name.com_.log** に同様のエラーが表示されることがあります。

```
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_get_user_done] (0x0040): s2n exop request failed.
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_get_user_done] (0x0040): s2n exop request failed.
(2022-05-22  5:00:13): [be[ad.example.com]] [ipa_s2n_exop_done] (0x0040): ldap_extended_operation result: Server is busy(51), Too many extdom instances running.
```

ipaExtdomMaxInstances オプションの値を設定することで、**ipa-extdom** インスタンスの最大数を調整できます。この値は、0 より大きく、ワーカースレッドの総数より小さい整数でなければなりません。

前提条件

- LDAP Directory Manager のパスワード

手順

1. ワーカースレッドの総数を取得します。
   -

```
# ldapsearch -xLLLD cn=directory\ manager -W -b cn=config -s base nsslapd-threadnumber
Enter LDAP Password:
dn: cn=config
nsslapd-threadnumber: 16
```

これは、**ipaExtdomMaxInstances** の現在の値が 13 であることを意味します。

2. インスタンスの最大数を調整します。この例では、値を 14 に変更します。

```
# ldapmodify -D "cn=directory manager" -W
dn: cn=ipa_extdom_extop,cn=plugins,cn=config
changetype: modify
replace: ipaExtdomMaxInstances
ipaExtdomMaxInstances: 14
```

3. **ipaExtdomMaxInstances** の現在の値を取得します。

```
# ldapsearch -xLLLD "cn=directory manager" -W -b "cn=ipa_extdom_extop,cn=plugins,cn=config" |grep ipaextdommaxinstances

Enter LDAP Password:

ipaextdommaxinstances: 14
```

4. IdMDirectory Server のパフォーマンスを監視し、改善されない場合は、この手順を繰り返して **ipaExtdomMaxInstances** 変数の値を調整します。

9.5. IDM-AD 信頼デプロイメント向けに IDM サービス SSSD の調整

この手順では、IdM クライアントの SSSD サービス設定にチューニングオプションを適用し、大規模な AD 環境から情報を取得する時の応答時間を短縮します。

前提条件

- **/etc/sssd/sssd.conf** 設定ファイルを編集するには、**root** パーミッションが必要です。

手順

1. キャッシュされていないログイン 1 回にかかる秒数を決定します。
   a. IdM クライアント **client.example.com** で SSSD キャッシュを削除します。

```
[root@client ~]# sss_cache -E
```

   b. **time** コマンドを使用して、AD ユーザーとしてログインにかかる時間を測定します。この例では、IdM クライアント **client.example.com** の AD ドメイン **ad.example.com** から **ad-user** として、同じホストにログインします。

```
[root@client ~]# time ssh ad-user@ad.example.com@client.example.com
```

   c. できるだけ早くパスワードを入力します。

```
Password:
Last login: Sat Jan 23 06:29:54 2021 from 10.0.2.15
[ad-user@ad.example.com@client ~]$
```

d. できるだけ早くログアウトして経過時間を表示します。この例では、キャッシュされていないログインに約 **9** 秒かかります。

```
[ad-user@ad.example.com@client /]$ exit
logout
Connection to client.example.com closed.

real 0m8.755s
user    0m0.017s
sys     0m0.013s
```

2. テキストエディターで **/etc/sssd/sssd.conf** 設定ファイルを開きます。

3. Active Directory ドメインの **[domain]** セクションに以下のオプションを追加します。**pam_id_timeout** オプションおよび **krb5_auth_timeout** オプションを、キャッシュを使用しないログインにかかる秒数に設定します。AD ドメインにドメインセクションがない場合は作成します。

```
[domain/example.com/ad.example.com]
krb5_auth_timeout = 9
ldap_deref_threshold = 0
...
```

4. **pam** セクションに次のオプションを追加します。

```
[pam]
pam_id_timeout = 9
```

5. サーバーの **/etc/sssd/sssd.conf** ファイルを保存して閉じます。

6. SSSD サービスを再起動して、設定の変更を読み込みます。

```
[root@client ~]# systemctl restart sssd
```

関連情報

- IdM-AD 信頼デプロイメント用の IdM サーバーおよびクライアントでの SSSD の調整オプション

9.6. TMPFS での SSSD キャッシュのマウント

SSSD (System Security Services Daemon) は、常に LDAP オブジェクトをキャッシュに書き込みます。これらの内部 SSSD トランザクションは、データをディスクに書き込みますが、これは Random-Access Memory (RAM) からの読み取りと書き込みよりもはるかに遅くなります。

このパフォーマンスを改善するには、RAM に SSSD キャッシュをマウントします。

留意事項

- SSSD キャッシュが RAM にある場合には、再起動後にキャッシュされた情報は保持されません。
- IdM サーバーの SSSD インスタンスは、同じホストの Directory Server への接続が切断されないので、IdM サーバーではこの変更を安全に実行できます。
- IdM クライアントでこの調整を実行し、IdM サーバーへの接続が切断されると、再起動後のユーザー認証は、接続を再確立するまでできません。

前提条件

- **/etc/fstab** 設定ファイルを編集するには、**root** 権限が必要になります。

手順

1. **tmpfs** 一時ファイルシステムを作成します。
   a. SSSD ユーザーが **config.ldb** ファイルを所有していることを確認します。

```
# ls -al /var/lib/sss/db/config.ldb
-rw-------. 1 sssd sssd 1286144 Jun  8 16:41 /var/lib/sss/db/config.ldb
```

   b. 次のエントリーを **/etc/fstab** ファイルに 1 行で追加します。

```
tmpfs /var/lib/sss/db/ tmpfs
size=300M,mode=0700,uid=sssd,gid=sssd,rootcontext=system_u:object_r:sssd_var_lib_
t:s0 0 0
```

   この例では、300MB キャッシュを作成します。IdM および AD ディレクトリーサイズにあわせて **size** を調整します (LDAP エントリー 1 万につき 100 MB 想定)。
2. 新しい SSSD キャッシュディレクトリーをマウントします。

```
[root@host ~]# mount /var/lib/sss/db/
```

3. SSSD を再起動して、この設定の変更を反映します。

```
[root@host ~]# systemctl restart sssd
```

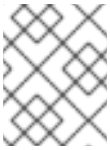
9.7. 大規模な IDM-AD 信頼デプロイメント用に IDM サーバーとクライアントを調整するための SSSD.CONF のオプション

/etc/sssd/sssd.conf 設定ファイルで次のオプションを使用して、IdM-AD の信頼が大規模にデプロイされている場合に、IdM サーバーおよびクライアントで SSSD のパフォーマンスを調整できます。

9.7.1. IdM サーバーのチューニングオプション

ignore_group_members

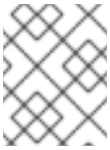
ユーザーの認証および承認時には、グループに所属する全ユーザーではなく、あるユーザーがどのグループに所属するかを把握することが重要です。**ignore_group_members** を **true** に設定すると、SSSD はメンバーではなくグループオブジェクトに関する情報のみを取得するので、パフォーマンスが大幅に向上します。



注記

id user@ad-domain.com コマンドはそのまま正しいグループリストを返しますが、**getent group** ad-group@ad-domain.com は空のリストを返します。

デフォルト値	**false**
推奨値	**true**



注記

デプロイメントで compat ツリーを持つ IdM サーバーがある場合は、このオプションを **true** に設定しないでください。

subdomain_inherit

subdomain_inherit オプションを使用すると、**ignore_group_members** の設定を、信頼できる AD ドメインの設定に適用できます。**subdomain_inherit** オプションに記載されている設定は、メイン (IdM) ドメインと AD サブドメインの両方に適用されます。

デフォルト値	**none**
推奨値	**subdomain_inherit = ignore_group_members**

9.7.2. IdM クライアントのチューニングオプション

pam_id_timeout

このパラメーターは、ID ルックアップ時にアイデンティティープロバイダーへのラウンドトリップが過剰になることを回避するために PAM セッションからの結果がキャッシュされる時間を制御します。デフォルト値の **5** 秒を使用する場合には、複雑なグループメンバーシップが IdM サーバーおよび IdM クライアント側で設定されていない環境では不十分な可能性があります。Red Hat は、キャッシュされていない場合の 1 回のログインにかかる秒数に、**pam_id_timeout** を設定することを推奨します。

デフォルト値	**5**
推奨値	**キャッシュされていないログインが 1 回にかかる秒数**

krb5_auth_timeout

krb5_auth_timeout を増やすと、ユーザーが多数のグループに所属する環境で、複雑なグループ情報を処理する時間数を増やすことができます。Red Hat は、キャッシュされていない場合の 1 回のログインにかかる秒数に、このタイムアウトの値を設定することを推奨します。

デフォルト値	**6**

推奨値	**キャッシュされていないログインが 1 回にかかる秒数**

ldap_deref_threshold

逆参照ルックアップを使用して、1回の LDAP 呼び出しで全グループメンバーを取得します。**ldap_deref_threshold** の値は、内部キャッシュに含まれないグループメンバー数を指定し、逆参照ルックアップをトリガーします。見つからないメンバーが少ない場合には、個別に検索します。大規模な環境では、逆参照ルックアップに時間がかかり、パフォーマンスが低下する可能性があります。逆参照検索を無効にするには、このオプションを **0** に設定します。

デフォルト値	**10**
推奨値	**0**

9.8. 関連情報

- 大規模な IdM-AD 信頼デプロイメント向けの SSSD のパフォーマンスチューニング

第10章 WSGI プロセスのチューニング

長時間実行される API プロセスが原因で要求が失敗する場合は、それらの API プロセスをチューニングすると効果が得られます。

デフォルトでは、IPA は 64 ビットシステム上の API サービスに Web Server Gateway Interface (WSGI) プロセスを 4 つ割り当てます。この 4 プロセスというデフォルトの制限は、メモリー節約のために実装されています。WSGI プロセスの数を増やすと、CPU 使用率とメモリー消費量は増加しますが、より多くの要求を受け入れることができます。デフォルトでは、IPA は WSGI プロセスごとに約 100 - 110 MB の常駐メモリーを API に使用します。これを推奨される上限である 16 プロセスに調整すると、消費量が約 1.3 GB になります。

手順

- **/etc/httpd/conf.d/ipa.conf** ファイルの processes 値を変更します。

```
WSGIDaemonProcess ipa processes=<4> threads=1 maximum-requests=500 \
```

長時間実行される API エンドポイントは、いずれもチューニングによる効果を得ることができます。このチューニングの決定はユーザーが行う必要があります。

たとえば、OpenStack インストールは、複数のサービスを含む複数のコントローラーで構成されています。各サービスは、すべての内部通信が Transport Layer Security (TLS) 経由で行われるように、証明書を要求します。コントローラーまたはコンピュートノードをインストールまたは更新するときに、これらの証明書が要求または更新されることがあります。複数のコントローラーまたはコンピュートノードが関係する状況では、証明書要求の量がかなり多くなることがあります。これらの要求は自動化されているため、同時またはほぼ同時に発生します。WSGI スレッドの数を増やすと、インストールを完了することができます。