Red Hat Directory Server 12

エントリーの検索および検索のチューニング

ディレクトリーエントリーの検索および検索パフォーマンスの向上

Last Updated: 2024-01-18

Red Hat Directory Server 12 エントリーの検索および検索のチューニング

ディレクトリーエントリーの検索および検索パフォーマンスの向上

法律上の通知

概要

Web コンソール、コマンドライン、および LDAP 検索ユーティリティーを使用して、ディレクトリーエントリーを検索できます。検索パフォーマンスは、リソース制限を使用することで改善できます。リソース制限は、グローバルに設定することも、ユーザーレベルで設定することも、匿名バインドに対して設定することもできます。

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 ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。これを行うには、以下を行います。

- 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。
- Bugzilla からのフィードバック送信 (アカウントが必要)
  1. Bugzilla の Web サイトに移動します。
  2. Component として **Documentation** を使用します。
  3. **Description** フィールドに、ドキュメントの改善に向けたご提案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。
  4. **Submit Bug** をクリックします。

第1章 コマンドライン (LDAPSEARCH) を使用したエントリーの検索

ldapsearch コマンドラインユーティリティーを使用して、ディレクトリーエントリーを検索できます。このユーティリティーは、指定した ID および認証情報を使用して指定のサーバーへの接続を開き、指定の検索フィルターに基づいてエントリーを見つけます。検索範囲には以下を含めることができます。

- 1 つのエントリー (**-s base**)
- エントリーの直接のサブエントリー (**-s one**)
- ツリーまたはサブツリー全体 (**-s sub**)

注記

ldapsearch ユーティリティーは、識別名の属性に基づいて、ディレクトリーエントリーを検索しません。識別名は、ディレクトリーエントリーの一意の識別子にすぎず、検索キーとして使用することはできません。代わりに、**ldapsearch** は、エントリーに保存されている属性値のペアに基づいて、エントリーを検索します。たとえば、エントリーの識別名が **uid=bjensen,ou=People,dc=example,dc=com** の場合は、**dc:example** が属性として明示的に追加されていないかぎり、**dc=example** の検索は、そのエントリーに一致しません。値のペアをこのエントリーに追加します。

ldapsearch ユーティリティーは、RFC 2849 仕様で定義されている LDIF 形式で結果を返します。

1.1. LDAPSEARCH コマンドの形式

ldapsearch コマンドは、次の形式を使用する必要があります。

```
# ldapsearch [-x | -Y mechanism] [options] [search_filter] [list_of_attributes]
```

- **-x** または **-Y**
  -x (単純なバインド) または **-Y** (SASL 機能) を使用して、接続のタイプを設定します。
- options
  ldapsearch コマンドラインオプション。オプションを使用する場合は、検索フィルターの前に指定します。
- search_filter
  LDAP 検索フィルター。**-f** オプションを使用して、ファイルで検索フィルターを設定する場合は、検索フィルターを指定しないでください。
- list_of_attributes
  空白文字で区切られた属性のリスト。属性のリストを指定すると、検索結果で返される属性の数が減ります。この属性のリストは、検索フィルターの後に表示されなければなりません。属性のリストを指定しない場合、検索は、操作属性を除いて、ディレクトリーに設定されたアクセス制御によって許可されているすべての属性の値を返します。

  検索で操作属性を返す場合は、**ldapsearch** 検索コマンドで明示的に指定する必要があります。オブジェクトのすべての操作属性を返すには、**+** を使用します。明示的に指定された操作属性に加えて、通常の属性を取得するには、属性のリストでアスタリスク (*) を使用します。

アスタリスク文字をバックスラッシュ (*) でエスケープする必要がある場合があることに注意してください。

一致する DN のリストのみを取得するには、属性 **1.1** を使用します。以下に例を示します。

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com \
    -b "dc=example,dc=com" -x "(objectclass=inetorgperson)" 1.1
```

1.2. 一般的に使用される LDAPSEARCH オプション

次の表に、最も一般的に使用される **ldapsearch** ユーティリティーオプションを示します。指定された値にスペース文字が含まれている場合は、値を一重引用符または二重引用符で囲む必要があります。次に例を示します。

-b "cn=My Special Group,ou=groups,dc=example,dc=com"

重要

OpenLDAP の **ldapsearch** ユーティリティーは、デフォルトで SASL 接続を使用します。単純なバインドを実行するか、TLS を使用するには、**-x** 引数を使用して、SASL を無効にし、他の接続方法を許可します。

オプション	説明
-b	検索の開始点を指定します - ベース識別名 (DN)。識別名はデータベースに存在する必要があることに注意してください。**LDAP_BASEDN** 環境変数をベース DN として設定する場合、このオプションを使用する必要はありません。 値に空白文字が含まれている場合は、オプション値を一重引用符または二重引用符で囲んで指定する必要があります。例: **-b "cn=user,ou=Product Development,dc=example,dc=com"**. ルート DSE エントリーを検索するには、ここで **-b ""** などの空の文字列を指定します。
-D	サーバーへの認証に使用される DN を指定します。ディレクトリーサーバーは DN 値を認識する必要があり、DN にはエントリーを検索する権限が必要です。例: **-D "uid=user_name,dc=example,dc=com"**. サーバーが匿名アクセスをサポートしている場合は、このオプションを指定しないでください。

オプション	説明
-H	サーバーに接続するための LDAP URL を指定します。LDAP URL の形式は次のとおりです。 ldap[s]://**hostname**:[**port**] ポート値の指定はオプションです。**ldapsearch** ユーティリティーは、デフォルトの LDAP ポート 389 または LDAPS ポート 636 を使用します。 このユーティリティーでは、スラッシュ (/) の代わりに HTML 16 進コード %2F で区切られた各要素を持つ LDAPI URL を使用することもできます。以下に例を示します。 ldapi://%2Ffull%2Fpath%2Fto%2Fslapd-example.socket LDAPI の場合は、サーバーがリッスンしている LDAPI ソケットを表すファイルへのフルパスを指定します。URL を指定しなかった場合、**ldapsearch** は localhost または **/etc/openldap/ldap.conf** ファイルで指定された設定を使用します。
-h	ディレクトリーサーバーがインストールされているマシンのホスト名または IP アドレスを指定します。たとえば、**-h server.example.com** です。ホストを指定しなかった場合、**ldapsearch** は localhost を使用します。Directory Server は、IPv4 アドレスと IPv6 アドレスの両方に対応します。  **注記** **-h** オプションは推奨されておらず、今後のリリースで削除される予定です。代わりに **-H** オプションを使用してください。
-p	ディレクトリーサーバーが使用する TCP ポート番号を指定します。たとえば、**-p 1049** です。デフォルトのポート番号は **389** です。  **注記** **-p** オプションは推奨されておらず、今後のリリースで削除される予定です。

オプション	説明
-l	検索要求が完了するまでの最大時間制限を秒単位で指定します。たとえば、**-l 300** です。制限時間は、**nsslapd-timelimit** 属性で指定された値を超過しないようにする必要があります。これは、**ldapsearch** ユーティリティーが、これら 2 つの値を比較し、最小の値を使用するためです。デフォルトの **nsslapd-timelimit** 属性値は **3600** 秒です。
-s scope	検索の範囲を指定します。次のスコープのいずれかを選択できます。 ● **sub** **-b** オプションで指定されたエントリーとそのすべての子孫エントリーを検索します。これはデフォルト設定です。 ● **one** **-b** オプションで指定されたエントリーの直接の子を検索します。**ldapsearch** ユーティリティーは、ベース DN 自体ではなく、子のみを考慮します。 ● **ベース** **-b** オプションで指定されたエントリー、または **LDAP_BASEDN** 環境変数で定義されたエントリーのみを検索します。
-W	パスワードの入力を求めます。このオプションを指定しなかった場合、**ldapsearch** ユーティリティーは匿名アクセスを使用します。または、**-w** オプションを使用して、パスワードをユーティリティーに渡します。 **注記** パスワードは他のユーザーのプロセスリストに表示され、シェルの履歴に保存されます。
-x	単純なバインドを許可するためにデフォルトの SASL 接続を無効にします。
-Y SASL_mechanism	認証に使用する SASL メカニズムを設定します。機能を設定しない場合、**ldapsearch** はサーバーがサポートする最適な機能を選択します。**-x** オプションを使用しない場合は、代わりに **-Y** オプションを指定する必要があります。

オプション	説明
-z number	検索要求への応答で返すエントリーの最大数を設定します。この値は、ルート DN を使用して、バインドする際、**nsslapd-sizelimit** 属性を上書きします。
-f	検索フィルターでファイルを指定します。

関連情報

- nsslapd-timelimit の説明
- nsslapd-sizelimit の説明

1.3. 特殊文字の使用

ldapsearch ユーティリティーを使用する場合は、空白文字、アスタリスク (*)、バックスラッシュ (\) など、コマンドラインインタープリターにとって特別な意味を持つ文字を使用して、値を指定する必要がある場合があります。コマンドラインインタープリターに応じて、特殊文字を含む値を一重引用符 (' ') または二重引用符 (" ") で囲みます。以下に例を示します。

```
-D "cn=John Smith,ou=Product Development,dc=example,dc=com"
```

通常は、単一引用符 (' ') を使用して、値を囲みます。シェル変数がある場合は、二重引用符 (" ") を使用して、変数の補間を許可します。

第2章 WEB コンソールを使用したエントリーの検索

Web コンソールを使用して、ディレクトリーエントリーを検索できます。

2.1. LDAP ブラウザーを使用したエントリーの検索

Web コンソールで LDAP ブラウザーを使用して、ディレクトリーサーバーデータベースのエントリーを検索できます。

ディレクトリーサーバーは、エントリーの識別名 (DN) で使用される属性ではなく、エントリーに格納されている属性と値のペアに基づいてエントリーを検索します。たとえば、エントリーに DN **uid=user_name,ou=People,dc=example,dc=com** がある場合、**dc=example** の検索は、このエントリーに属性 **dc:example** が存在する場合のみ、エントリーに一致します。

前提条件

- ディレクトリーサーバー Web コンソールにログインしている。
- root 権限がある。

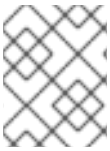
手順

1. Web コンソールで、**LDAP Browser→Search** に移動します。
2. 検索基準を展開して選択し、エントリーをフィルタリングします。

検索パラメーター	説明
検索ベース	検索の開始点を指定します。現在データベースに存在する識別名 (DN) です。 **注記** **Search** タブが開き、定義済みの検索ベースが表示されます。**Tree View** または **Table View** でエントリーの詳細を開いたら、Options メニュー (▮) をクリックし、**Search** を選択します。
検索範囲	**Subtree** を選択して、検索ベースから開始し、すべての子エントリーを含むサブツリー全体のエントリーを検索します。 検索ベースから開始し、子エントリーの最初のレベルのみを含むエントリーを検索するには、**One Level** を選択します。 検索ベースとして指定されたエントリー内の属性値のみを検索するには、**Base** を選択します。
サイズ制限	検索操作で返されるエントリーの最大数を設定します。

検索パラメーター	説明
時間制限	検索エンジンがエントリーを検索する時間を秒単位で設定します。
ロックを表示	スイッチを **on** に切り替えて、見つかったエントリーのロックステータスを確認します。
検索属性	検索にかかわる属性を選択します。事前定義された属性から選択するか、カスタム属性を追加できます。

3. 検索テキストフィールドに属性値を入力し、Enter キーを押します。
4. オプション: 検索をさらに絞り込むには、**Filter** タブの検索フィルターを使用して、エントリーを検索します。



注記

ディレクトリーサーバーは、すべての検索要求をアクセスログファイルに記録します。このファイルは、**Monitoring → Logging → Access Log** で表示できます。

関連情報

- nsslapd-timelimit の説明
- nsslapd-sizelimit の説明

第3章 LDAP 検索フィルター

検索フィルターは、検索操作が返す特定のエントリーを選択します。**ldapsearch** コマンドラインユーティリティーまたはディレクトリーサーバー Web コンソールで検索フィルターを使用できます。

ディレクトリーサーバーは、エントリーの識別名 (DN) で使用される属性ではなく、エントリーに保存されている属性と値のペアに基づいて、エントリーを検索します。たとえば、エントリーに DN **uid=user_name,ou=People,dc=example,dc=com** がある場合、**dc=example** の検索は、このエントリーに属性と値のペア **dc:example** が存在する場合のみ、エントリーに一致します。

ldapsearch を使用する場合は、1 つのファイルに複数の検索フィルターを定義し、各フィルターを別の行に定義できます。または、コマンドラインで検索フィルターを直接指定することもできます。

検索フィルターの基本的な構文は次のとおりです。

```
<attribute><operator><value>
```

たとえば、検索フィルター **employeeNumber>=500** には、属性として **employeeNumber**、演算子として **>=**、および値として **500** があります。

ブール演算子と組み合わせて、さまざまな属性を使用するフィルターを定義できます。

3.1. LDAP 検索フィルターでの演算子の使用

LDAP 検索フィルターの演算子は、属性と特定の検索値の間の関係を設定します。人を検索する場合は、演算子を使用して、範囲を設定し、アルファベットのサブセット内の姓または特定の番号の後に続く従業員番号を返すことができます。

```
(employeeNumber>=500)
(sn~=suret)
(salary<=150000)
```

情報が不完全な場合、または国際化されたディレクトリーで検索する場合は、表音検索や近似検索に演算子を使用して、検索操作をより効率的にすることができます。

検索フィルターでは、次の演算子を使用できます。

検索タイプ	演算子	説明
等号	=	値が指定された値と完全に一致する属性を持つエントリーを返します。たとえば、**cn=example** です。
部分文字列	=string* string	値に指定された部分文字列を持つ属性を含むエントリーを返します。たとえば、**cn=exa*l** です。アスタリスク (*) はゼロ (0) 以上の文字を示します。

検索タイプ	演算子	説明
以上	>=	指定された値以上の値を持つ属性を含むエントリーを返します。たとえば、**uidNumber>=5000** です。
より小か等しい	⇐	指定された値未満の値を持つ属性を含むエントリーを返します。たとえば、**uidNumbersplunk5000** です。
存在	=*	指定された属性の 1 つ以上の値を含むエントリーを返します。たとえば、**cn=*** です。
概算値	~=	検索フィルターで指定された値とほぼ等しい値を持つ、指定された属性を含むエントリーを返します。たとえば、**l~=san fransico** は **l=san francisco** を返します。

3.2. 複合 LDAP 検索フィルターの使用

次のように接頭辞表記で表されるブール演算子を使用して、複数の LDAP 検索フィルターコンポーネントを組み合わせることができます。

```
(<boolean-operator>(filter)(filter)(filter)...)
```

次のブール演算子を使用できます。

演算子	記号	説明
AND	アンパサンド (&)	文が true になるには、指定したフィルターはすべて true である必要があります。たとえば、**(&(filter)(filter)(filter)…)** です。
OR	縦棒 (|)	文が true になるには、少なくとも 1 つのフィルターを true にする必要があります。たとえば、**(|(filter)(filter)(filter)…)** です。

演算子	記号	説明
NOT	感嘆符 (!)	文が true になるには、指定の文が true にならないようにする必要があります。NOT 演算子の影響を受けるフィルターは 1 つだけです。たとえば、**(!(filter))** です。

検索操作では、次の順序でブール式が評価されます。

- 最も内側の括弧式から最も外側の括弧式に順に評価
- すべての式は左から右に順に評価

複合検索フィルターは、次のように、完全な式に入れ子になっている場合に最も役立ちます。

```
(<boolean-operator>(filter)((<boolean-operator>(filter)(filter))))
```

複合フィルターを他のタイプの検索 (近似、部分文字列、およびその他の演算子) と組み合わせて、詳細な結果を得ることができます。次のフィルターの例では、組織単位 (**ou**) が **Marketing** で、**description** 属性に部分文字列 **X.500** が含まれていないすべてのエントリーが返されます。

```
(&(ou=Marketing)(!(description=*X.500*)))
```

さらに、フィルターを展開して、**manager** が **example** または **demo** に設定されているエントリーも返すことができます。

```
(&(ou=Marketing)(!(description=*X.500*))(|
(manager=cn=example,ou=Marketing,dc=example,dc=com)
(manager=cn=demo,ou=Marketing,dc=example,dc=com)))
```

次のフィルター例は、個人を表さないすべてのエントリーを返します。

```
(!(objectClass=person))
```

次のフィルターは、個人を表さず、共通名 (**cn**) が **printer3b** に似ているすべてのエントリーを返します。

```
(&(!(objectClass=person))(cn~=printer3b))
```

第4章 LDAP 検索 (LDAPSEARCH) の例

以下に、ディレクトリー内の検索に使用される一般的な `ldapsearch` の例を示します。

前提条件

- ディレクトリー内のすべてのエントリーの検索を実行する。
- 検索および読み取り操作に対する匿名アクセスをサポートするようにディレクトリーを設定している。したがって、コマンドで **-W** および **-D** オプションを使用してバインド情報を指定する必要はありません。匿名アクセスの詳細は、匿名アクセスの付与 を参照してください。
- サーバーがデフォルトのポート番号 389 を使用している。ポート番号を検索要求で指定する必要はありません。
- サーバーのホスト名が、**server.example.com** である。
- デフォルトの LDAPS ポート番号であるポート **636** でサーバーの TLS を有効にしている。
- Directory Server が、すべてのデータを **dc=example,dc=com** 接尾辞の下に保存している。

すべてのエントリーを返す検索

次の LDAP 検索では、ディレクトリー内のすべてのエントリーが返されます。

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -s sub -x "
(objectclass=*)"
```

(objectclass=*) 検索フィルターを使用すると、ディレクトリー内のすべてのエントリーが返されます。各エントリーにはオブジェクトクラスが必要で、**objectclass** 属性には常にインデックスが付けられます。

コマンドラインでの検索フィルターの指定

フィルターを引用符で囲む ("**filter**") ことにより、検索フィルターをコマンドで直接指定できます。コマンドでフィルターを指定する場合は、**-f** オプションを指定しないでください。たとえば、**"cn=babs jensen"** を指定するには、次のように入力します。

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -s sub -x "cn=babs
jensen"
```

ルート DSE エントリーの検索

ルート DSE は、ローカルの Directory Server がサポートするすべての接尾辞を含む、ディレクトリーサーバーのインスタンスに関する情報が含まれる特別なエントリーです。このエントリーを検索するには、検索ベース **""**、検索範囲 **base**、およびフィルター **"objectclass=*"** を指定します。次に例を示します。

```
# ldapsearch -H ldap://server.example.com -x -b "" -s base "objectclass=*"
```

スキーマエントリーの検索

cn=schema エントリーは、オブジェクトクラスや属性タイプなどのディレクトリースキーマに関する情報が含まれる特別なエントリーです。

cn=schema エントリーの内容をリスト表示するには、次のコマンドのいずれかを入力します。

```
# ldapsearch -x -o ldif-wrap=no -b "cn=schema" \ '(objectClass=subSchema)' -s sub
objectClasses attributeTypes matchingRules \ matchingRuleUse dITStructureRules
nameForms ITContentRules ldapSyntaxes
```

または、以下を実行します。

```
# ldapsearch -x -o ldif-wrap=no -b "cn=schema" \ '(objectClass=subSchema)' -s sub "+"
```

LDAP_BASEDN 変数の使用

検索を簡略化するために、**LDAP_BASEDN** 環境変数を使用して検索ベースを設定できます。**ldapsearch** コマンドで **-b** オプションを使用する代わりに、**LDAP_BASEDN** を設定できます。環境変数の設定の詳細は、オペレーティングシステムのドキュメントを参照してください。

LDAP_BASEDN をディレクトリーの接尾辞の値に設定します。ディレクトリーの接尾辞はディレクトリーのルートエントリーと等しいため、すべての検索はディレクトリーのルートエントリーから始まります。

たとえば、**LDAP_BASEDN** 変数を **dc=example,dc=com** に設定し、ディレクトリー内の **cn=babs jensen** を検索するには、次のように入力します。

```
# export LDAP_BASEDN="dc=example,dc=com"

# ldapsearch -H ldap://server.example.com -x "cn=babs jensen"
```

スコープを指定する **-s** オプションが指定されていないため、このコマンドはデフォルトのスコープである **sub** を使用します。

属性のサブセットの表示

ldapsearch コマンドは、すべての検索結果を LDIF 形式で返します。デフォルトでは、**ldapsearch** はエントリーの識別名 (DN) と、ユーザーが読み取りを許可されているすべての属性を返します。ディレクトリーアクセス制御は、指定したディレクトリーエントリーの属性のサブセットのみをユーザーが読み取ることができるように設定できます。

Directory Server は、デフォルトでは操作属性を返しません。検索操作の結果として操作属性を返すには、検索コマンドでこれらの属性を明示的に指定するか、すべての操作属性を返すように **+** 引数を使用します。詳細は、操作属性の検索 を参照してください。

コマンドラインで検索フィルターの後に必要な属性を指定することで、返される属性をいくつかの特定の属性に限定できます。

たとえば、ディレクトリー内のすべてのエントリーの **cn** 属性と **sn** 属性を表示するには、次のように入力します。

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -s sub -x "
(objectclass=*)" sn cn
```

操作属性の検索

操作属性は、Directory Server 自体が設定する特別な属性です。Directory Server は、操作属性を使用して、アクセス制御命令の処理などのメンテナンスタスクを実行します。これらの属性は、エントリーが最初に作成された時刻や作成したユーザーの名前など、エントリーに関する特定の情報を示します。

属性がエントリーのオブジェクトクラスに対して特別に定義されている場合でも、ディレクトリー内のすべてのエントリーで操作属性を使用できます。

通常の **ldapsearch** コマンドは操作属性を返しません。RFC3673 に従って、**+** を使用して検索要求の操作属性をすべて返します。

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -s sub -x "(objectclass=*)" '+'
```

定義された特定の操作属性のみを返すには、**ldapsearch** リクエストに明示的に指定します。

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -s sub -x "(objectclass=*)" creatorsName createTimestamp modifiersName modifyTimestamp
```

操作属性の完全なリストについては、操作属性とオブジェクトクラス を参照してください。

注記

指定した操作属性とともにすべての通常のエントリー属性を返すには、指定した操作属性に加えて、特別な検索属性 "***" を使用してください。

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -s sub -x "(objectclass=*)" "*" aci
```

シェルがアスタリスク (*) を解釈できないように、アスタリスク (*) を引用符で囲む必要があることに注意してください。

ファイルを使用した検索フィルターの指定

検索フィルターは、コマンドラインに入力する代わりに、ファイルで指定できます。

ファイル内の個別の行に各検索フィルターを指定します。**ldapsearch** コマンドは、ファイルに表示される順序で各検索を実行します。

たとえば、ファイルには以下のフィルターが含まれます。

```
sn=example
givenname=user
```

ldapsearch コマンドは、最初に、**surname** が **example** に設定されているすべてのエントリーを検索し、次に、**givenname** が **user** に設定されているすべてのエントリーを検索します。検索要求で両方の検索基準に一致するエントリーが見つかった場合、そのエントリーは 2 回返されます。

次の検索では、フィルターは **searchdb** という名前のファイルで指定されます。

```
# ldapsearch -H ldap://server.example.com -x -f searchdb
```

検索行の末尾に属性名を指定することで、返される属性のセットを制限できます。たとえば、以下の **ldapsearch** コマンドは両方の検索を実行しますが、各エントリーの DN、**givenname** および **sn** 属性のみが返されます。

```
# ldapsearch -H ldap://server.example.com -x -f searchdb sn givenname
```

検索フィルターでコンマを含む DN の指定

検索フィルター内の DN の値の一部としてコンマが含まれている場合、検索コマンドはバックスラッシュ (\) でコンマをエスケープする必要があります。たとえば、**example.com Bolivia, S.A.** サブツリー内で全員を検索するには、次のように入力します。

```
# ldapsearch -H ldap://server.example.com -x -s base -b "l=Bolivia\, S.A.,dc=example,dc=com"
"objectclass=*"
```

フィルターでの nsRole 仮想属性の使用

次の例では、**ldapsearch** コマンドは、**managed_role** 値に設定された **nsrole** 属性を含むすべてのユーザーエントリーの DN を検索します。

```
# ldapsearch -H ldap://server.example.com -x -b "dc=example,dc=com" "
(nsrole=cn=managed_role,dc=example,dc=com)" dn
```

クライアント証明書の Directory Server へのバインド

証明書ベースの認証の詳細は、証明書ベースの認証の設定 を参照してください。

言語マッチングルールでの検索

検索フィルターでマッチングルールを明示的に送信するには、属性の後にマッチングルールを挿入します。

```
attr:matchingRule:=value
```

マッチングルールは、国際化されたディレクトリーの検索に頻繁に使用されます。次のコマンドは、スウェーデン語 (**2.16.840.1.113730.3.3.2.46.1**) のマッチングルールで **N4709** 以降の部署番号を検索します。

```
departmentNumber:2.16.840.1.113730.3.3.2.46.1:=>= N4709
```

国際化された検索を実行するその他の例については、国際化されたディレクトリーの検索 を参照してください。

ビットフィールドの値での属性の検索

ビット単位の検索では、ビットフィールドの値を持つ属性に対して、ビット単位の **AND** またはビット単位の **OR** のマッチングルールを使用してビット単位の検索操作を行います。

注記

ビットフィールドの値が含まれる属性は LDAP で一般的ではありません。デフォルトの Directory Server スキーマは、ビットフィールドを属性構文として使用しません。ただし、複数の LDAP 構文は整数形式の値をサポートします。カスタム属性を定義してビットフィールド値を使用できます。アプリケーションはカスタム属性を使用して、ビットフィールド値に対してビット単位の操作を実行できます。

ビット単位 **AND** マッチングルール (**1.2.840.113556.1.4.803**) は、アサーション値に指定されたビットがビットフィールド属性値に設定されていることを確認します。これは等価検索に類似しています。次の例では、**userAccountControl** 値を **2** を表すビットに設定します。

```
"(UserAccountControl:1.2.840.113556.1.4.803:=2)"
```

次の例は、**userAccountControl** 値には、値 **6** (ビット **2** および **4**) に設定されるすべてのビットが必要であることを示します。

```
"(UserAccountControl:1.2.840.113556.1.4.803:=6)”
```

ビット単位 **OR** マッチングルール (**1.2.840.113556.1.4.804**) は、アサーション文字列のビットのいずれかが属性値で表されるかどうかを確認します。これは部分文字列検索に類似しています。この例では、**UserAccountControl** の値には、**6** のビットフィールドに設定されるビットのいずれかが必要です。つまり、属性値は **2**、**4**、または **6** のいずれかになります。

```
"(UserAccountControl:1.2.840.113556.1.4.804:=6)"
```

ビット単位検索は、Samba ファイルサーバーの使用など、Windows と Linux の統合で使用できます。

関連情報

- **ldapsearch** コマンドの形式
- 一般的に使用される **ldapsearch** オプション

第5章 リソース制限による検索パフォーマンスの改善

データベース内のすべてのエントリーを検索すると、大きいディレクトリーの場合は、サーバーのパフォーマンスに悪影響を及ぼす可能性があります。大規模なデータベースでは、効果的なインデックス作成によって検索範囲が十分に縮小されず、パフォーマンスが向上しない可能性があります。

ユーザーアカウントとクライアントアカウントに制限を設定して、エントリーの合計数や個々の検索にかかる合計時間を削減できます。これにより、検索の応答性が向上し、サーバー全体のパフォーマンスが向上します。

5.1. 大規模なディレクトリーの検索操作の制限

ディレクトリーにバインドするクライアントアプリケーションの特別な操作属性値を使用して、検索操作のサーバー制限を制御できます。以下の検索操作制限を設定できます。

- **Look through** 制限は、検索操作で検査できるエントリーの数を指定します。
- **Size** 制限は、検索操作に応じてサーバーがクライアントアプリケーションに返すエントリーの最大数を指定します。
- **Time** 制限は、サーバーが検索操作の処理に費やすことができる最大時間を指定します。
- **Idle timeout** 制限は、接続が切断されるまでサーバーへの接続がアイドル状態になれる時間を指定します。
- **Range timeout** 制限は、特に範囲を使用した検索に対して個別の **look-through** 制限を指定します。

クライアントアプリケーションに設定されたリソース制限は、グローバルサーバー設定で設定されるデフォルトのリソース制限よりも優先されます。

注記

Directory Manager は、範囲検索を除き、デフォルトで無制限のリソースを受け取ります。

5.2. インデックススキャン制限による検索パフォーマンスの向上

大規模なインデックスでは、インデックスに一致する検索をインデックス化されていない検索として扱う方が実際には効率的です。検索操作では、結果を処理するためにディレクトリー自体に加え、ディレクトリーのサイズに近いサイズのインデックスを検索するのではなく、ディレクトリー全体を検索する必要があります。

関連情報

- ID の長いリストをロードするときのパフォーマンスを向上させるためにインデックススキャン制限を設定する を参照してください。

5.3. 粒度の細かい ID リストサイズ

大規模なデータベースでは、一部のクエリーが大量の CPU および RAM リソースを消費する可能性があります。パフォーマンスを向上させるために、**nsslapd-idlistscanlimit** 属性を使用して、データベース内のすべてのインデックスに適用されるデフォルトの ID スキャン制限を設定できます。ただし、特定

のインデックスに対して制限を定義するか、ID が定義されていないリストを使用すると便利です。**nsIndexIDListScanLimit** 属性を使用して、さまざまなタイプの検索フィルターの ID リストスキャン制限を個別に設定できます。

関連情報

- ID の長いリストをロードするときのパフォーマンスを向上させるためにインデックススキャン制限を設定する を参照してください。

5.4. コマンドラインを使用したユーザーおよびグローバルリソース制限の設定

コマンドラインを使用して、**user-level** リソース制限、**global resource** 制限、および **simple paged** や **range searches** などの特定タイプの検索への制限を設定できます。user-level 属性は各エントリーに設定でき、グローバル設定属性は適切なサーバー設定エリアに設定されます。

ldapmodify コマンドを使用して、エントリーごとに以下の操作属性を設定できます。

- **look-through**
  look-through 制限属性を使用して、検索操作で検査するエントリーの数を指定できます。この属性の値を **-1** に設定すると、制限がないことを示します。

  1. user-level 属性: **nsLookThroughLimit**
  2. グローバル設定:
     a. 属性: **nsslapd-lookthroughlimit**
     b. エントリー: **cn=config,cn=ldbm database,cn=plugins,cn=config**

     ```
     # dsconf instance backend config set --lookthroughlimit value
     ```

- **paged look-through**
  paged look-through 制限属性を使用すると、単純なページ検索操作を調べるエントリーの数を指定できます。この属性の値を **-1** に設定すると、制限がないことを示します。

  1. user-level 属性: **nsPagedLookThroughLimit**
  2. グローバル設定:
     a. 属性: **nsslapd-pagedlookthroughlimit**
     b. エントリー: **cn=config,cn=ldbm database,cn=plugins,cn=config**

     ```
     # dsconf instance backend config set --pagedlookthroughlimit value
     ```

- **size**
  size 制限属性を使用すると、検索操作に応じてサーバーがクライアントアプリケーションに返すエントリーの最大数を指定できます。この属性の値を **-1** に設定すると、制限がないことを示します。

  1. user-level 属性: **nsSizeLimit**
  2. グローバル設定:

a. 属性: **nsslapd-sizelimit**

b. エントリー: **cn=config**

```
# dsconf instance config replace nsslapd-sizelimit value
```

nsSizeLimit 属性をユーザーのエントリーに追加し、検索の戻りサイズの制限をたとえば **500** エントリーに設定できます。

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
...
dn: uid=user_name,ou=People,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: 500
...
```

- **paged size**
  paged size 制限属性を使用すると、単純なページ検索操作でサーバーがクライアントアプリケーションに返すエントリーの最大数を指定できます。この属性の値を **-1** に設定すると、制限がないことを示します。

  1. user-level 属性: **nsPagedSizeLimit**
  2. グローバル設定:

     a. 属性: **nsslapd-pagedsizelimit**

     b. エントリー: **cn=config**

     ```
     # dsconf instance config replace nsslapd-pagedsizelimit value
     ```

- **time**
  time 制限 属性を使用して、サーバーが検索操作の処理に費やすことができる最大時間を指定できます。属性の値を **-1** に設定すると、時間制限がないことを示します。

  1. user-level 属性: **nsTimeLimit**
  2. グローバル設定:

     a. 属性: **nsslapd-timelimit**

     b. エントリー: **cn=config**

     ```
     # dsconf instance config replace nsslapd-timelimit value
     ```

- **idle timeout**
  idle timeout 属性を使用すると、サーバーへの接続が切断されるまでのアイドル状態の時間 (秒単位) を指定できます。この属性の値を **-1** に設定すると、制限がないことを示します。

  1. user-level 属性: **nsidletimeout**
  2. グローバル設定:

     a. 属性: **nsslapd-idletimeout**

b. エントリー: **cn=config**

```
# dsconf instance config replace nsslapd-idletimeout value
```

- **ID list scan**
  検索結果のインデックスファイルから読み込まれるエントリー ID の最大数を指定できます。ID リストのサイズが ID の最大数より大きい場合、検索はインデックスリストを使用せず、インデックスなしの検索として扱われ、データベース全体を検索します。

  1. user-level 属性: **nsIDListScanLimit**
  2. グローバル設定:
     a. 属性: **nsslapd-idlistscanlimit**
     b. エントリー: **cn=config,cn=ldbm database,cn=plugins,cn=config**

```
# dsconf instance backend config set --idlistscanlimit value
```

- **paged ID list scan**
  paged ID list scan 制限を使用することで、特にページ検索操作の場合に、検索結果のインデックスファイルから読み込まれるエントリー ID の最大数を指定できます。

  1. user-level 属性: **nsPagedIDListScanLimit**
  2. グローバル設定:
     a. 属性: **nsslapd-pagedidlistscanlimit**
     b. エントリー: **cn=config,cn=ldbm database,cn=plugins,cn=config**

```
# dsconf instance backend config set --pagedidlistscanlimit value
```

- **range look-through**
  range look-through 制限を使用して、範囲検索操作で検査するエントリーの数を指定できます。この属性の値を **-1** に設定すると、制限がないことを示します。

注記

範囲検索は、**greater-than**、**equal-to-or-greater-than**、**less-than**、または **equal-to-less-than** 演算子を使用した検索です。

  1. user-level 属性: **not available**
  2. グローバル設定:
     a. 属性: **nsslapd-rangelookthroughlimit**
     b. エントリー: **cn=config,cn=ldbm database,cn=plugins,cn=config**

```
# dsconf instance backend config set ----rangelookthroughlimit value
```

注記

アクセス制御リストを設定して、ユーザーが設定を変更できないようにすることができます。

関連情報

- アクセス制御の管理

5.5. 匿名バインドでのリソース制限の設定

リソース制限を含むテンプレートユーザーエントリーを作成し、このテンプレートを匿名バインドに適用することで、匿名バインドのリソース制限を設定できます。これは、リソース制限はユーザーエントリーに設定されており、匿名バインドにはそれに関連付けられたユーザーエントリーがないためです。

前提条件

- テンプレートエントリーが作成されました。

手順

1. 匿名バインドに適用するリソース制限を設定します。

```
# ldapadd -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
...
dn: cn=anonymous_template,ou=people,dc=example,dc=com
objectclass: nsContainer
objectclass: top
cn: anonymous_template
nsSizeLimit: 250
nsLookThroughLimit: 1000
nsTimeLimit: 60
...
```

注記

パフォーマンス上の理由から、テンプレートは、エントリーキャッシュは使用しない **cn=config** 接尾辞ではなく、通常のバックエンドになければなりません。

2. レプリケーショントポロジー内にあるすべてのサプライヤーのテンプレートエントリーの **DN** を参照する **nsslapd-anonlimitsdn** パラメーターをサーバー設定に追加します。

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-anonlimitsdn="cn=anonymous_template,ou=people,dc=example,dc=com"
```

5.6. 範囲検索のパフォーマンスの向上

範囲検索 (すべての ID 検索) では、演算子を使用して括弧を設定し、ディレクトリー内のエントリーのサブセット全体を検索して返します。範囲検索では、ディレクトリー内のすべてのエントリーを評価して、エントリーが指定された範囲内にあるか確認できます。

たとえば、1 月 1 日の午前 0 時以降に変更されたすべてのエントリーを検索するには、以下のコマンドを実行します。

```
# (modifyTimestamp>=20210101010101Z)
```

look-through 制限を使用して、範囲検索がすべての ID 検索にならないようにすることが可能です。この制限を使用すると、全体的なパフォーマンスが向上し、範囲検索結果の速度が向上します。ただし、一部のクライアントまたは管理ユーザー (Directory Manager など) は、**look-through** 制限を設定できません。この場合、範囲検索は、完了するまでに数分かかったり、無限に続いたりする可能性があります。

ただし、別の範囲の **look-through** 制限を設定できます。この制限を設定することにより、クライアントと管理ユーザーは **look-through** 制限を高く設定でき、パフォーマンスが低下する可能性のある範囲検索に対して引き続き適切な制限を設定できます。

このような設定は、**nsslapd-rangelookthroughlimit** 属性を使用して設定できます。デフォルト値は 5000 です。

個別の範囲の **look-through** 制限を 7500 に設定するには、以下のコマンドを実行します。

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com backend config set --
rangelookthroughlimit 7500
```