opentext™

OpenText™ eDirectory CE 24.4 (v9.3)

Optimierungshandbuch

Januar 2025

Rechtliche Hinweise

Copyright 2009–2024 Open Text Alle Rechte vorbehalten.

Für Produkte und Services von Open Text oder seinen verbundenen Unternehmen und Lizenznehmern („Open Text“) gelten nur die Gewährleistungen, die in den Gewährleistungserklärungen, die solchen Produkten beiliegen, ausdrücklich beschrieben sind. Aus den in dieser Publikation enthaltenen Informationen ergibt sich keine zusätzliche Gewährleistung. Open Text haftet nicht für technische oder redaktionelle Fehler oder Auslassungen in diesem Dokument. Die in diesem Dokument enthaltenen Informationen sind vorbehaltlich etwaiger Änderungen.

Inhalt

Info zu diesem Handbuch und zur Bibliothek **5**

1 Überblick **7**

Voraussetzungen . . . 7

2 OpenText eDirectory-Subsysteme **9**

FLAIM-Datenbank . . . 9
- Checkpoint . . . 10
- Indizes . . . 10
- Rollforward-Protokoll (Einzeltransaktionsprotokoll) . . . 11
- Attributcontainer in FLAIM . . . 11

Thread-Pool . . . 12

3 Analyse der Systemengpässe **15**

Festplatten-E/A-Subsystem . . . 15
Prozessorsubsystem . . . 16
Arbeitsspeichersubsystem . . . 17
Netzwerksubsystem . . . 17

4 Optimieren der OpenText eDirectory-Subsysteme **19**

FLAIM-Datenbank . . . 19
- Auswahl der Indexe . . . 20
- Optimierung für Aktualisierungen . . . 21

Thread-Pool . . . 21
ACLs . . . 21
- Verbessern von Such- und Lesevorgängen in OpenText eDirectory . . . 22
- Deaktivieren von ACL-Vorlagen . . . 23

Reproduktion . . . 25
Solid State Disk (SSD) . . . 26
NMAS-Anmeldeaktualisierungsintervall . . . 26
SSL-Overhead . . . 26
Import/Konvertierung/Export (Import Convert Export, ICE) . . . 26
ldif2dib . . . 26
Größere NCP-Pakete . . . 27

5 OpenText eDirectory-Konfiguration **29**

Konfigurieren des FLAIM-Subsystems . . . 29
- Cache-Hardlimit . . . 29
- Dynamische Anpassung der Grenze . . . 29

Ändern der FLAIM-Cache-Einstellungen . . . 29
- Ändern der FLAIM-Cache-Einstellungen über OpenText iMonitor . . . 30

Ändern der FLAIM-Cache-Einstellungen über _ndsdb.ini 31

Info zu diesem Handbuch und zur Bibliothek

Dieses Handbuch beschreibt, wie OpenText™ eDirectory CE 24.4 (v9.3) mittels Analyse und Feinabstimmung optimiert werden kann, um in allen Bereitstellungen eine bessere Produktleistung zu erzielen.

Die neueste Version des *OpenText™ eDirectory-Optimierungshandbuchs* finden Sie auf der OpenText eDirectory-Onlinedokumentationsseite (https://www.netiq.com/documentation/edirectory-93/).

Zielgruppe

Dieses Handbuch richtet sich an Netzwerkadministratoren.

Weitere Informationen in der Bibliothek

Die Bibliothek enthält folgende Informationsressourcen:

XDASv2-Administrationshandbuch

Beschreibt die Konfiguration und Arbeit mit XDASv2 zur Prüfung von OpenText eDirectory und OpenText Identity Manager.

Installationsanleitung

Beschreibt die Installation von OpenText eDirectory. Das Handbuch richtet sich an Netzwerkadministratoren.

Verwaltungshandbuch

Beschreibt die Verwaltung und Konfiguration von OpenText eDirectory.

Handbuch zur Fehlersuche

Beschreibt die Behebung von Problemen bei OpenText eDirectory.

Handbuch zu neuen Funktionen

Beschreibt die neuen Funktionen von OpenText eDirectory.

Diese Handbücher sind auf der OpenText eDirectory -Dokumentationswebsite (https://www.netiq.com/documentation/edirectory-93/) verfügbar.

Informationen zur eDirectory-Verwaltungsfunktion finden Sie im OpenText Identity Console Administration Guide (https://www.netiq.com/documentation/identity-console/) (OpenText Identity Console-Administrationshandbuch).

1 Überblick

OpenText eDirectory 9.3 ist eine normenkonforme, plattformübergreifende, extrem skalierbare, fehlertolerante und leistungsfähige Verzeichnisdienstlösung. Dieses Handbuch enthält Informationen dazu, wie Sie die OpenText eDirectory-Umgebung zur Leistungsverbesserung optimieren können.

Die Leistungsoptimierung ist eine komplexe Aktivität. Sie erfordert ein gutes Verständnis von OpenText eDirectory und der Subsysteme des Betriebssystems. Sie beruht auf der Überwachung des Systems, um Engpässe zu identifizieren und nacheinander zu beheben. Oft stehen nur beschränkte Ressourcen zur Verfügung und die Optimierung beschränkt sich auf OpenText eDirectory und das Betriebssystem.

Lesen Sie in diesem Handbuch zunächst den Abschnitt Voraussetzungen, bevor Sie Optimierungsmaßnahmen ausführen. Fahren Sie dann mit den weiteren Abschnitten fort. Das Kapitel OpenText eDirectory-Subsysteme beschreibt die primären Subsysteme, die einen Einfluss auf die Leistung von OpenText eDirectory haben. Das Kapitel Analyse der Systemengpässe beschreibt verschiedene Systemressourcen und ihren Einfluss auf die Leistung von OpenText eDirectory. Im Kapitel Optimieren der OpenText eDirectory-Subsysteme wird beschrieben, wie OpenText eDirectory analysiert und unter verschiedenen Bedingungen und in verschiedenen Umgebungen optimiert werden kann. Das Kapitel OpenText eDirectory-Konfiguration beschreibt die Konfiguration verschiedener optimierbarer Parameter.

Voraussetzungen

Stellen Sie sicher, dass die folgenden allgemeinen Voraussetzungen erfüllt sind, bevor Sie das System mit der Zielsetzung einer Leistungsverbesserung anpassen:

- Eine gute OpenText eDirectory-Baumstruktur kann die Leistung von eDirectory verbessern. Beachten Sie hierzu Folgendes:
  - Anwendungen lesen alle Informationen lokal auf dem Server, ohne Anfragen aneinanderketten zu müssen.
  - OpenText eDirectory verarbeitet die Objektreferenzen automatisch und effizient. Sofern möglich, sollten Objekte auf einem Server sich nicht auf Objekte beziehen, die nicht lokal auf dem Server vorhanden sind, da die Erhaltung von Referenzen zu nichtlokalen Objekten mehr Zeit in Anspruch nehmen kann. Wenn solche Referenzen vorhanden sind, müssen Backlinks gepflegt werden. In großen Bereitstellungen wird dies umständlich.
  - Wenn Sie eine Gruppe mit 10.000 Mitgliedern oder mehr benötigen, empfehlen sich dynamische Gruppen. Dadurch kann der Overhead reduziert werden, der mit der Erhaltung von Referenzen für eine so große Anzahl an Personen verbunden ist. Wählen Sie die Konfiguration der dynamischen Gruppen sorgfältig aus. Die Verwendung mehrerer dynamischer Gruppen mit ungeeigneten Suchkriterien kann zu einer Überlastung des Servers führen und die allgemeine Serverleistung beeinträchtigen. Wenn Suchvorgänge zu lange dauern, ist möglicherweise der gewählte Index nicht wirksam. Minimieren Sie die Verwendung von regulären (statischen) Gruppen, da dies das Browsen im Baum bei der Anmeldung erhöhen kann.

- Achten Sie auf eine effiziente Nutzung von ACLs. Verwenden Sie beispielsweise den Trustee [This] und weisen Sie ihn auf der Container-Ebene zu, statt eine ACL-Vorlage zu verwenden, die sich selbst Rechte zuweist. Je weniger ACLs Sie verwenden, desto besser ist die Leistung. Weitere Informationen zu ACLs finden Sie in „OpenText eDirectory Rights“ (OpenText eDirectory-Rechte) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3)-Administrationshandbuch).
- Verteilen Sie die Last auf mehrere Reproduktionsserver.
- Auch wenn das Browsen im Baum durch eine gute Baumstruktur reduziert wird, ist es dennoch manchmal erforderlich. Sie können erwägen, den Algorithmus „Advanced Referral Costing“ zu verwenden, der im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3) eDirectory-Administrationshandbuch) beschrieben ist.
- Wenn Anmeldungen lange dauern, können Sie die Anmeldeaktualisierungen deaktivieren. Für NDS- und NetIQ Modular Authentication Service (NMAS)-Anmeldungen stehen verschiedene Methoden zur Deaktivierung der Anmeldeaktualisierungen zur Verfügung. Sie sollten sich jedoch unbedingt der Auswirkungen auf die Sicherheit (http://www.novell.com/documentation/nmas33/admin/data/bg8dphs.html) bewusst sein.

- Führen Sie Zustandsprüfungen über iMonitor aus. Weitere Informationen finden Sie in „Viewing OpenText eDirectory Server Health“ (OpenText eDirectory-Serverzustand anzeigen) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3) eDirectory-Administrationshandbuch). Sie müssen Folgendes sicher stellen:
  - Die Uhrzeit ist auf allen Reproduktionsservern synchronisiert.
  - Die Reproduktionssynchronisierung und Hintergrundprozesse sind in einem guten Zustand.

2 OpenText eDirectory-Subsysteme

Dieser Abschnitt behandelt die OpenText eDirectory-Subsysteme.

- „FLAIM-Datenbank“, auf Seite 9
- „Thread-Pool“, auf Seite 12

FLAIM-Datenbank

OpenText eDirectory verwendet FLAIM als Datenbank. FLAIM (Flexible Adaptable Information Manager) wird für herkömmliche, flüchtige und komplexe Informationen verwendet. Es handelt sich um eine äußerst skalierbare Datenbank-Engine, die ein Nebenläufigkeitsmodell für mehrere Leser und einen einzelnen Schreiber unterstützt. Die Leser blockieren die Schreiber nicht, und die Schreiber blockieren nicht die Leser.

Physisch organisiert FLAIM die Daten in Blöcken. Einige Blöcke werden üblicherweise im Arbeitsspeicher behalten. Sie stellen den Block-Cache dar. Der Eintrags-Cache (manchmal Datensatz-Cache genannt) speichert logische Einträge der Datenbank im Cache. Aus den Elementen im Block-Cache werden Einträge konstruiert. FLAIM pflegt Hash-Tabellen für beide Caches. Die Größe des Hash-Bucket wird regelmäßig auf Grundlage der Elementanzahl angepasst.

Standardmäßig arbeitet OpenText eDirectory mit einer Blockgröße von 4 KB. Die Größe des Block-Cache für das Caching der gesamten DIB entspricht der DIB-Größe. Der Eintrags-Cache muss etwa zwei- bis viermal so große sein wie die DIB.

Beim Abrufen eines Eintrags sucht FLAIM zunächst im Eintrags-Cache nach dem Eintrag. Wenn der Eintrag vorhanden ist, ist kein Lesen aus dem Block-Cache erforderlich. Beim Abrufen eines Blocks von der Festplatte sucht FLAIM zuerst im Cache nach dem Block. Wenn der Block vorhanden ist, ist kein Lesevorgang auf dem Datenträger erforderlich.

Beim Hinzufügen oder Ändern eines Eintrags wird für die entsprechenden Blocks des Eintrag auf dem Datenträger nicht direkt ein Commit ausgeführt. Daher sind der Datenträger und der Arbeitsspeicher unter Umständen nicht synchronisiert. Die am Eintrag vorgenommenen Aktualisierungen werden jedoch im Rollforward-Protokoll (Einzeltransaktionsprotokoll) (RFL) protokolliert. Das RFL dient dem Wiederherstellen von Transaktionen nach einem Systemfehler.

LRU (Least Recently Used) ist der Algorithmus zum Ersetzen von Einträgen im Cache.

- „Checkpoint“, auf Seite 10
- „Indizes“, auf Seite 10
- „Rollforward-Protokoll (Einzeltransaktionsprotokoll)“, auf Seite 11
- „Attributcontainer in FLAIM“, auf Seite 11

Checkpoint

Anhand eines Checkpoints wird die Datenträgerversion der Datenbank auf den gleichen kohärenten Zustand wie die Datenbank des Arbeitsspeichers (im Cache gespeichert) gebracht. FLAIM kann während der minimalen Aktualisierungsaktivität an der Datenbank einen Checkpoint ausführen. Der Prozess wird jede Sekunde ausgeführt und schreibt die veränderten Blocks (dirty Cache) auf den Datenträger. Im Cache geänderte, aber noch nicht auf den Datenträger geschriebene Blocks werden als „dirty Blocks“ (veränderte Blocks) bezeichnet. FLAIM ruft eine Sperre der Datenbank ab und führt die größtmögliche Menge Arbeit aus, bis entweder der Checkpoint abgeschlossen ist oder ein anderer Thread darauf wartet, die Datenbank zu aktualisieren. Um zu verhindern, dass die Datenbank auf dem Datenträger zu stark desynchronisiert wird, wird unter bestimmten Bedingungen ein Checkpoint erzwungen, auch wenn Threads auf die Aktualisierung der Datenbank warten:

- Wenn der Checkpoint-Thread einen Checkpoint nicht innerhalb eines festgelegten Zeitintervalls (standardmäßig 3 Minuten) abschließen kann, wird der Checkpoint erzwungen und der veränderte Cache (dirty Cache) wird bereinigt.
- Wenn die Größe des veränderten Cache den Wert `maxdirtycache` übersteigt (sofern festgelegt), wird ein Checkpoint erzwungen, der die Größe des veränderten Cache auf `mindirtycache` (sofern festgelegt) bzw. auf null reduziert.

Indizes

Ein Index ist ein Satz Schlüssel, deren Anordnung die Suche nach einem bestimmten Schlüssel im Index deutlich beschleunigt. Indexschlüssel werden erstellt, indem die Inhalte eines oder mehrerer Felder (Attribute) von den Einträgen extrahiert werden. Die Indexe werden im Block-Cache gepflegt. Änderungen an den indizierten Attributen erfordern eine Änderung der Indexblöcke.

OpenText eDirectory definiert einen standardmäßigen Indexsatz für Systemattribute (Felder). Systemattribute wie `parentID` oder `ancestorID` werden für Suchen in einer Ebene und in Teilbäumen verwendet. Diese Indexe können nicht aufgehoben oder gelöscht werden. Sie werden intern im Verzeichnis verwendet. Standardmäßige Indexe werden für Attribute wie CN, `Nachname`, `Vorname` usw. verwendet. Indexe können vom Typ „Präsenz“, „Wert“ oder „Teilzeichenkette“ sein. Diese Indexe können aufgehoben werden. Nach einem Löschen werden sie automatisch neu erstellt.

Indexe können mit OpenText Identity Console oder mit dem Lightweight Directory Access Protocol (LDAP)-Dienstprogramm „ndsindex“ erstellt werden. Indexe (http://www.netiq.com/documentation/edir83/) sind serverspezifisch.

Durch Aktivieren der Storage Manager (`StrMan`)-Kennung in DSTrace (ndstrace), können Sie den für die Suchabfragen gewählten Index anzeigen.

Folgendes Beispiel betrifft ein DSTrace-Protokoll für eine Teilbaumsuche mit „`cn=admin`", CN.

```
3019918240 StrMan: Iter #b239c18 query ((Flags&1)==1) &&
((CN$217A$.Flags&8=="admin") && (AncestorID==32821))

3019918240 StrMan: Iter #b239c18 index = CN$IX$220
```

Folgendes Beispiel betrifft ein DSTrace-Protokoll für eine Teilbaumsuche mit „`Description= This is for testing`", `AncestorID`.

```
2902035360 StrMan: Iter #83075b0 query ((Flags&1)==1) &&
((Description$225A$.Flags&8=="This is for testing") &&
(AncestorID==32821))

2902035360 StrMan: Iter #83075b0 index = AncestorID_IX
```

Um die Suchleistung mit der serverseitigen Sortierung zu verbessern, verwenden Sie die Option -a, um dem Attribut `AncestorID` als Präfix einen Wert aus der Liste der Attribute zu geben, die beim Erstellen eines neues Index übermittelt werden.

Die folgende Tabelle zeigt den Trend der Erhöhung der DIB-Größe beim Erstellen einfacher/zusammengesetzter Indizes für eine bestimmte Anzahl von Objekten. Wir empfehlen, beim Erstellen einfacher/zusammengesetzter Indizes auf diese Testdaten Bezug zu nehmen:

Anzahl der Objekte	Erhöhung der DIB-Größe mit einfachem Index, der eine Attributgröße von 20 Zeichen enthält	Erhöhung der DIB-Größe mit einfachem Index, der eine Attributgröße von 200 Zeichen enthält	Erhöhung der DIB-Größe mit einem zusammengesetzten Index, der eine Attributgröße von 200 Zeichen enthält
23 Tsd.	24 KB	3 MB	7 MB
47 Tsd.	50 KB	6 MB	15 MB
0,5 Millionen	4 MB	55 MB	100 MB
1 Million	9 MB	111 MB	210 MB

Rollforward-Protokoll (Einzeltransaktionsprotokoll)

FLAIM protokolliert die Vorgänge für jede Aktualisierungstransaktion in einer Rollforward-Protokolldatei (Einzeltransaktionsprotokolldatei) (RFL). Ein RFL dient der Wiederherstellung von Transaktionen nach einem Systemfehler oder der Wiederherstellung von einer Sicherung. Die RFL-Datei wird nach jedem Abschließen eines Checkpoints abgeschnitten, sofern sie nicht aktiviert ist (`rflkeepfiles`), indem eine interaktive dauerhafte Sicherung (http://www.netiq.com/documentation/edir93/) eingesetzt wird.

Attributcontainer in FLAIM

Zur optimalen Auslastung des Eintrags-Cache und zur besseren Attributsuche speichert FLAIM Attribute, die längere Werte oder eine größere Anzahl davon enthalten, separat in einem Attributcontainer (Attribute Container). Standardmäßig werden die Attribute automatisch in den Container verschoben, wenn für das Attribut Folgendes gilt:

- Es hat mehr als 25 Werte.
- Einer seiner Werte ist größer als 2.048 Byte.

Um das automatische Verschieben von Attributen in Container zu deaktivieren, fügen Sie in der `_ndsdb.ini`-Datei `disablemovetoattrcontainer=1` hinzu und starten Sie OpenText eDirectory neu.

Mit OpenText eDirectory können Sie das Verschieben von Attributen jetzt flexibel planen. Zunächst lassen Sie sich die verschiebbaren Attribute anzeigen, dann können Sie nach eigenem Ermessen deren Verschieben planen.

Mit dem Befehl `ndscheck` zeigen Sie die Anzahl der Attribute an, die in Attributcontainer verschoben werden können. Um die Details von Attributen anzuzeigen, verwenden Sie das iMonitor-Attribut **dsContainerReadyAttrs** für die Pseudo-Serverobjekte in der **Agentenkonfiguration**. Der Benutzer sieht diese Attribute, die zur Indexierung gekennzeichnet sind, auch im Agentenzustand.

Starten Sie die Containerisierung der Attribute des Pseudo-Server-Objekts mithilfe des Reparaturbefehls für Einzelobjekte: `ndsrepair`. Zum Verschieben eines Attributs erweitern Sie den Befehl `ndsrepair` um den neuen erweiterten Schalter `-am`, gefolgt vom Namen des Attributs (siehe nachstehend), und führen Sie ihn aus.

```
ndsrepair –J <Pseudo server object ID> –Ad –AM/–am <attribute name>
```

Sobald ein Attribut in den Attributcontainer verschoben wurde, erstellt OpenText eDirectory einen Systemindex, der den Namen des Attributs enthält. Ein einmal verschobenes Attribut kann nicht wieder in seinen Ursprungscontainer verschoben werden.

HINWEIS: Wenn ein Attribut einen Wert hat, der größer als 2.048 Byte ist, erfolgt dennoch das Verschieben in den Container, aber OpenText eDirectory erstellt keinen Systemindex.

Thread-Pool

Aus Gründen der Leistungsverbesserung arbeitet OpenText eDirectory mit mehreren Threads. Beim Arbeiten mit mehreren Threads werden bei ausgelastetem System weitere Threads zur Verarbeitung der Last erstellt. Einige Threads werden beendet, um zusätzlichen Overhead zu vermeiden. Ein häufiges Erstellen und Vernichten von Threads ist ineffizient und kostenintensiv. Statt für jede Aufgabe neue Threads zu erzeugen und wieder zu vernichten, werden mehrere Threads gestartet und in einem Pool organisiert. Das System weist die Threads aus dem Thread-Pool je nach Bedarf verschiedenen Aufgaben zu. Die Aufgaben werden in zwei Arten von Warteschlangen gehalten:

- Aufgaben, die eine sofortige Planung erfordern, werden in die Warteschlange „Bereit“ gestellt.
- Aufgaben, die zu einem späteren Zeitpunkt geplant werden müssen, werden in die Warteschlange „Wartet“ gestellt.

Der Thread-Pool wird nicht von allen Modulen verwendet. Die tatsächliche Anzahl der Threads für den Prozess übersteigt die im Thread-Pool vorhandene Anzahl. Beispielsweise verwaltet FLAIM die eigenen Hintergrund-Threads separat.

Durch Ausführen des Befehls `ndstrace -c threads` werden folgende Poolstatistiken zurückgegeben:

- Gesamtzahl der erzeugten, beendeten und ruhenden Threads.
- Gesamtzahl der aktuellen Worker-Threads und höchste Anzahl der Worker-Threads.
- Anzahl der Aufgaben und höchste Anzahl der Aufgaben in der Warteschlange „Bereit“.
- Mindest-, Höchst- und Durchschnittsdauer in der Warteschlange „Bereit“ in Mikrosekunden.
- Aktuelle Anzahl und Höchstzahl der Aufgaben in der Warteschlange „Wartet“.

Beispiel für einen Thread-Pool:

```
Thread Pool Information
Summary      : Spawned 42, Died 5
Pool Workers : Idle 8, Total 37, Peak 37
Ready Work   : Current 0, Peak 10, maxWait 67436 us
Sched delay  : Min 14 us, Max 1052004 us, Avg: 792 us
Waiting Work : Current 17, Peak 21
```

Folgende Parameter sind für Thread-Pools verfügbar:

- **n4u.server.max-threads:** höchste Anzahl an Threads, die in einem Pool verfügbar sein können.
- **n4u.server.idle-threads:** höchste Anzahl an ruhenden Threads, die in einem Pool verfügbar sein können.
- **n4u.server.start-threads:** Anzahl der gestarteten Threads.

Führen Sie die Befehle `ndsconfig get` und `ndsconfig set` aus, um die Größe des Thread-Pools abzurufen bzw. festzulegen.

3 Analyse der Systemengpässe

Verschiedene Systemressourcen beeinflussen die Leistung von OpenText eDirectory. Durch das Aufrüsten auf die neueste Version des Betriebssystems kann die Leistung verbessert werden.

- „Festplatten-E/A-Subsystem“, auf Seite 15
- „Prozessorsubsystem“, auf Seite 16
- „Arbeitsspeichersubsystem“, auf Seite 17
- „Netzwerksubsystem“, auf Seite 17

Festplatten-E/A-Subsystem

Das Festplatten-Subsystem stellt den häufigsten Engpass dar. Der E/A nimmt bei längeren Warteschlangen eine recht lange Zeit in Anspruch, was zu einer höheren Festplattenauslastung und leerlaufenden Prozessurzyklen führt. Mit dem iostat-Werkzeug können Sie während erwarteter Spitzenlasten Angaben zur durchschnittlichen Antwortzeit ermitteln.

Lese-, Schreib- und Aktualisierungsvorgänge auf der Festplatte können sequenziell oder mit zufälligem Zugriffsmuster erfolgen. In OpenText eDirectory-Bereitstellungen wird für Lese- und Schreibvorgänge üblicherweise ein zufälliges Zugriffsmuster verwendet.

Einige Lösungen für Arbeitslasten mit zufälligem Muster:

- Arbeitsspeicher erhöhen. Dadurch können häufig verwendete Daten bzw. im Voraus gelesene Daten auf der Dateisystemebene im Cache gespeichert werden. Außerdem ermöglicht es ein Caching der DIB im FLAIM-Subsystem.
- Dedizierte Volumes für die DIB verwenden. Die Dateisystemleistung steigt bei Volumes, die näher am Spindel erstellt werden. Dedizierte Volumes für das RFL und andere Protokolle verwenden.
- Wenn die Datenträger im Laufe der Zeit aufgrund der größeren Fragmentierung längere Latenzzeiten aufweisen, sollte eine Defragmentierung ausgeführt werden.
- Getrennte Laufwerke für das FLAIM-RFL hinzufügen. Diese Art der Protokollierung kann auf Hochgeschwindigkeitsfestplatten ausgeführt werden.
- RAID 10 (1+0)-Umgebung mit mehreren Laufwerken verwenden.

Die von OpenText eDirectory erstellten Dateien können eine Größe bis zu 4 GB erreichen. Dateisysteme, die zur Verarbeitung größerer Dateien optimiert sind, arbeiten mit OpenText eDirectory besonders effizient.

- Für Solaris™ ist das Veritas* VxFS-Dateisystem ein erweiterungsbasiertes Dateisystem, bei dem die Metadaten des Dateisystems für die Verarbeitung großer Dateien optimiert sind. Das UFS-Dateisystem ist indirekt blockbasiert. Die Metadaten des Dateisystems werden in einer größeren Anzahl Blocks gespeichert. Die Metadaten können für größere Dateien sogar verteilt sein. UFS ist daher für größere Dateien langsamer.
- Für Linux™ stellt das Reiser-Dateisystem ein schnelles Journaling-Dateisystem dar. Es bietet bei großen DIB-Sätzen eine bessere Leistung als das ext3-Dateisystem. Der Rückschreibe-Journaling-Modus von ext3 bietet jedoch in etwa die gleiche Leistung wie das Reiser-Dateisystem, obwohl der standardmäßig geordnete Modus eine bessere Datenkonsistenz bietet. XFS ist ein leistungsfähiges Journaling-Dateisystem, das große Dateien verarbeiten kann und reibungslöse Datenübertragungen bietet. OpenText eDirectory 9.3 wird auf Plattformen mit der 32- oder 64-Bit-Version von SLES 15 mit XFS-Dateisystem unterstützt.

FLAIM unterstützt eine Blockgröße von 4 KB und 8 KB. Standardmäßig sind 4 KB festgelegt. Dies entspricht der standardmäßigen Blockgröße unter Linux (`tune2fs -l device`). Unter Solaris wird das UFS-Dateisystem jedoch mit einer standardmäßigen Blockgröße von 8 KB erstellt (`df -g mountpoint`). Wenn die FLAIM-Blockgröße kleiner ist als die Blockgröße des Dateisystems, können partielle Blockschreibevorgänge auftreten. Wenn die Datenbankblockgröße größer als die Blockgröße des Dateisystems ist, werden einzelne Blockschreibe- und Blocklesevorgänge in mehrere einzelne physische E/A-Operationen aufgeteilt. Daher sollten Sie darauf achten, dass die FLAIM-Blockgröße und die Blockgröße des Dateisystems übereinstimmen.

Die Blockgrößen können nur beim Erstellen der DIB gesteuert werden. Fügen Sie die Zeile „blocksize=8192“ zur Datei `_ndsdb.ini` hinzu, um die DIB mit einer Blockgröße von 8 KB zu erstellen.

Die richtige Blockgröße hängt von der durchschnittlichen Größe des FLAIM-Datensatzes in Ihrer Bereitstellung ab. Zur Ermittlung der richtigen Blockgröße für Ihre Bereitstellung sind empirische Tests mit geeigneten Prüfdaten erforderlich.

Prozessorsubsystem

OpenText eDirectory basiert auf einer extrem skalierbaren Architektur. Durch eine größere Zahl Prozessoren kann die Leistung verbessert werden. Bei hoher Belastung kann mindestens bis zum 12. Prozessor eine Leistungssteigerung beobachtet werden. Die Leistungssteigerung ist jedoch abhängig von der Leistung anderer Ressourcen bei steigender Last im System. Server sind häufig mit unzureichenden Festplatten und Speichern konfiguriert. Fügen Sie nur unter folgenden Bedingungen weitere Prozessoren hinzu:

- Wenn die durchschnittliche Last der aktuell verwendeten Prozessoren eine Auslastung von 75 % übertrifft. Wenn die aktuelle Prozessorauslastung unter 75 % liegt, kann die Leistung durch Hinzufügen weiterer Prozessoren möglicherweise nicht verbessert werden.
- Wenn das Hinzufügen von Prozessoren zu einer angemessenen Leistungssteigerung führt.

Wenn OpenText eDirectory mit zu vielen Threads konfiguriert ist, wird eine beträchtliche Prozessorarbeitszeit für das Umschalten zwischen Kontexten verwendet. In diesem Fall kann eine Reduzierung der Threads zu einem gesteigerten Durchsatz führen.

Arbeitsspeichersubsystem

Durch eine Vergrößerung des Arbeitsspeichers kann die Leistung von Serveranwendungen deutlich verbessert werden. Durch Speichern der OpenText eDirectory-Datenbank im Cache des Dateisystems bzw. im FLAIM-Cache kann die Leistung von Such- und Änderungsvorgängen gesteigert werden. In großen Bereitstellungen ist es jedoch nicht möglich, die gesamte DIB im Cache zu speichern. Vermeiden Sie das Auslagern von Seiten, auch wenn dies mit einer Reduzierung der FLAIM-Datensatzgröße und der Größe des Block-Cache verbunden ist. Mit dem vmstat-Werkzeug können Sie weitere Informationen zum Arbeitsspeichersubsystem ermitteln.

Da OpenText eDirectory den Arbeitsspeicher verwendet, benötigt jeder Thread des Thread-Pools 1 MB Arbeitsspeicher für seinen eigenen Stack. Standardmäßig ist die Größe des FLAIM-Cache auf 200 MB festgelegt.

Beim Start von OpenText eDirectory werden verschiedene ladbare Module gestartet. Die Architektur der ladbaren Module in OpenText eDirectory ermöglicht Ihnen jedoch, die Auswirkungen des Prozesses auf den Arbeitsspeicher zu reduzieren, indem nicht verwendete Module (z. B. SecretStore, LDAP oder eMBox) nicht geladen werden. Außerdem verfügen bestimmte Produkte wie IDM über Module, die intern in OpenText eDirectory ausgeführt werden.

Es kann so aussehen, als ob der von OpenText eDirectory verwendete Arbeitsspeicher ständig wächst. Dies liegt daran, dass der von einem OpenText eDirectory-Prozess freigegebene Arbeitsspeicher nicht zum Systempool zurückgegeben wird, weil der intern von OpenText eDirectory verwendete Speicherverwalter versucht, die Speicherzuweisungen für zukünftige Prozesse zu optimieren. Unter anderem aus diesem Grund wird eine dynamische Konfiguration von FLAIM nicht empfohlen. Mit dem Top-Werkzeug können Sie die ungefähre Größe des virtuellen Arbeitsspeichers des ndsd-Prozesses in der Bereitstellung ermitteln.

Der maximal zu einem Prozess zuweisbare Arbeitsspeicher ist auf verschiedene Weisen begrenzt. Eine bestimmte Menge Arbeitsspeicher wird vom Betriebssystem und anderen Prozessen im System verwendet. Das Betriebssystem kann den von einem Prozess verwendeten physischen Arbeitsspeicher beschränken.

Netzwerksubsystem

Üblicherweise steht in Bereitstellungen eine ausreichende Bandweite zur Verfügung, um Netzwerkspitzenlasten zu verarbeiten. Eine angemessen Bandweite reduziert Fehler, Kollisionen und verlorene Pakete. Mit dem netstat-Werkzeug können Sie Netzwerkstatistiken ermitteln.

Einige Betriebssysteme bieten anpassbare TCP/IP-Parameter zur Optimierung netzwerkintensiver Server. Weitere Informationen finden Sie in der Dokumentation des Betriebssystems.

Wenn das Netzwerk den Engpass darstellt, sollten Sie die Bandweite erhöhen. Auch die Konfiguration eines dedizierten privaten Netzwerks zwischen den Anwendungsservern und dem OpenText eDirectory-Server kann dazu beitragen, die Netzwerküberlastung zu senken.

4 Optimieren der OpenText eDirectory-Subsysteme

Dieser Abschnitt enthält folgende Informationen:

- „FLAIM-Datenbank“, auf Seite 19
- „Thread-Pool“, auf Seite 21
- „ACLs“, auf Seite 21
- „Reproduktion“, auf Seite 25
- „Solid State Disk (SSD)“, auf Seite 26
- „NMAS-Anmeldeaktualisierungsintervall“, auf Seite 26
- „SSL-Overhead“, auf Seite 26
- „Import/Konvertierung/Export (Import Convert Export, ICE)“, auf Seite 26
- „ldif2dib“, auf Seite 26
- „Größere NCP-Pakete“, auf Seite 27

FLAIM-Datenbank

Die Cache-Größe ist wohl der Faktor, der die Gesamtleistung von OpenText eDirectory am meisten beeinflusst. Je größer die Anzahl der Elemente (Blöcke und Einträge), die im Cache gespeichert werden können, desto besser ist die Gesamtleistung. Der Prozentsatz der im Cache gefundenen Blöcke oder Einträge wird als Trefferquote bezeichnet. Eine höhere Quote führt zu einer besseren Leistung. Die Trefferquote kann mit OpenText iMonitor angezeigt werden.

Der Block-Cache eignet sich am besten für Aktualisierungsoperationen. Der Eintrags-Cache eignet sich am besten für Vorgänge, die eine Suche nach einem Eintrag auf Basisebene ausführen. Für Suchen auf einer Ebene und für Suchen in Teilbäumen werden jedoch sowohl der Eintrags-Cache als auch der Block-Cache verwendet. Der Block-Cache wird zum Abrufen von Indexen verwendet. Erstellen Sie je nach Bedarf die richtige Art von Index. Weitere Informationen hierzu finden Sie unter „Auswahl der Indexe“, auf Seite 20.

Ein Fehler im Block-Cache kann zu einem Lesevorgang auf der Festplatte führen. Lesevorgänge auf der Festplatte sind ressourcenintensiv, können jedoch verhindert werden, indem ein Block vom Cache des Dateisystems abgerufen wird.

Die zum Speichern der gesamten Datenbank im Block-Cache erforderliche Menge Arbeitsspeicher entspricht nahezu der Größe der Datenbank auf der Festplatte. Zum Speichern der vollständigen Datenbank im Eintrags-Cache ist ein Arbeitsspeicher erforderlich, der zwei- bis viermal so groß ist wie die Datenbank auf der Festplatte. Wenn im System weniger Arbeitsspeicher verfügbar ist, versuchen Sie es mit einem kleineren Eintrags-Cache und einem erheblich größeren Block- oder Dateisystem-Cache.

Wenn die Lesevorgänge auf einen bestimmten Eintragssatz im Verzeichnis beschränkt sind, sollten Sie den Eintrags-Cache vergrößern, solange dies zu einer besseren Trefferquote für den Eintrags-Cache führt.

Wenn das Lesemuster zufällig ist und die DIB deutlich größer als der verfügbare Arbeitsspeicher, sollte der Block-Cache oder der Cache des Dateisystems größer sein als der Eintrags-Cache.

Die zur Leistungsoptimierung von OpenText eDirectory eingesetzten Methoden müssen stets empirisch getestet werden. Ein gutes Verhältnis zwischen Eintrags- und Block-Cache in suchintensiven Umgebungen ist 2:1. Vergewissern Sie sich, dass ausreichend Arbeitsspeicher für andere Prozesse übrig bleibt. Vermeiden Sie das Auslagern von Seiten, auch wenn dies mit einer Reduzierung der Größe des FLAIM-Cache verbunden ist.

Da FLAIM vorher zugewiesenes Caching bietet, wird der Speicher, der dem OpenText eDirectory-Cache zugewiesen ist, nie vom nativen Speicherverwalter des Betriebssystems fragmentiert.

Auswahl der Indexe

Mit Indexen kann die Leistung für Suchen, die auf eine Ebene oder einen Teilbaum bezogen sind, verbessert werden. Auch dynamische Gruppen nutzen Suchen, die auf eine Ebene bzw. einen Teilbaum bezogen sind. Indexe werden nicht für Suchen auf Basisebene verwendet.

Da ein Index vom Typ „Präsenz“ nicht zwischen vorhandenen und nicht vorhandenen (gelöschten) Werten differenziert, wird diese Art Index hauptsächlich für interne Zwecke eingesetzt. Wenn Anwendungen eine Suchabfrage vom Typ „Präsenz“ ausführen, wird diese Art Index nie verwendet. Für Anwendungen sollten daher keine Indexe vom Typ „Präsenz“ erstellt werden.

Anwendungen können einen Index vom Typ „Wert“ erstellen, was für die meisten Suchen ausreichend ist. FLAIM kann einen Index vom Typ „Wert“ zum Ausführen von Präsenz- und Teilzeichenkettensuchen über die Attribute verwenden.

Ein Index vom Typ „Teilzeichenkette“ kann die an einem Attribut ausgeführten Aktualisierungen deutlich verzögern. Die Anzahl der für die Unterstützung eines Teilzeichenkettenindexes erforderlichen Indexblöcke ist im Vergleich zum Index vom Typ „Wert“ recht hoch. Dies bedeutet, dass für das Speichern im Cache hier ein größerer Block-Cache erforderlich ist. Erstellen Sie einen Teilzeichenkettenindex nur, wenn dies erforderlich ist. Für die meisten Suchen ist ein Index vom Typ „Wert“ ausreichend. Wenn jedoch bei Teilzeichenkettensuchen mit einem Index vom Typ „Wert“ keine angemessene Leistung erzielt wird, kann für diese Attribute ein Teilzeichenkettenindex erstellt werden.

Wenn ein Suchvorgang trotz des gewählten Index lange dauert, können Sie einen neueren Index vom Typ „Wert“ für eines der Attribute des Suchfilters einführen. Wählen Sie das Attribut aus, für das die Indexerstellung das beste Ergebnis liefert.

Optimierung für Aktualisierungen

Der Block-Cache eignet sich am besten für Aktualisierungsoperationen. Indexe sind auch im Block-Cache vorhanden. Indexe tragen zu schnelleren Suchen bei. Zu viele Indexe jedoch führen dazu, dass der Server mit ihrer Pflege ausgelastet ist. Indexe werden geändert, wenn Attributwerte geändert, hinzugefügt oder gelöscht werden. Bei umfangreichen Upload-Vorgängen können Indexe deaktiviert werden, um das Heraufladen zu beschleunigen.

Wenn sich das RFL-Verzeichnis und das DIB-Verzeichnis auf verschiedenen Datenträgern befinden, wird eine bessere Leistung erzielt.

Über den Parameter `maxdirtycache` kann die zulässige Grenze für die Antwortzeit bei einem Aktualisierungsvorgang gesteuert werden. Wenn als annehmbarer Grenzwert für die Serverantwort beispielsweise 5 Sekunden definiert wird und die Schreibgeschwindigkeit auf dem Datenträger im Zufallszugriffsmodus 20 MB pro Sekdunde beträgt, sollte `maxdirtycache` auf den Wert 20x5 = 100 MB festgelegt werden. Vergewissern Sie sich, dass der Block-Cache über ausreichend Platz für diese veränderten („dirty“) Blöcke verfügt. Weitere Informationen finden Sie in „Ändern der FLAIM-Cache-Einstellungen über _ndsdb.ini“, auf Seite 31.

Thread-Pool

Standardmäßig können in einem Thread-Pool maximal 256 Threads verfügbar sein. Diese Zahl ist für die meisten Bereitstellungen ausreichend. In großen Bereitstellungen kann die Anzahl auf 512 Threads erhöht werden. In den folgenden Fällen empfiehlt es sich, die Anzahl der Threads im Pool zu erhöhen:

- Wenn die Anzahl der ruhenden Threads oft null beträgt.
- Wenn die durchschnittliche Zeit einer Aufgabe in der Warteschlange „Bereit“ hoch ist und steigt.
- Wenn die Anzahl der Aufgaben in der Warteschlange „Bereit“ hoch ist und steigt.

Erhöhen Sie die maximale Thread-Anzahl, solange dies die Leistung des Servers steigert. Das Erhöhen der maximalen Thread-Anzahl führt auch zu einer höheren Prozessorauslastung.

Weitere Informationen zum Anzeigen von Statistiken zum Thread-Pool finden Sie in „Viewing the Thread Pools Statistics“ (Anzeigen von Thread-Pool-Statistiken) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3)-Administrationshandbuch).

ACLs

- „Verbessern von Such- und Lesevorgängen in OpenText eDirectory“, auf Seite 22
- „Deaktivieren von ACL-Vorlagen“, auf Seite 23

Verbessern von Such- und Lesevorgängen in OpenText eDirectory

Eine LDAP-Suche in OpenText eDirectory gibt Ergebnisse je nach der Anzahl der für einen Benutzer zurückgegebenen Attribute (`inetOrgPerson`) zurück.

Wenn in OpenText eDirectory ein Objekt erstellt wird, können standardmäßige ACLs zum Objekt hinzugefügt werden. Dies ist abhängig von den ACL-Vorlagen in der Schemadefinition der Objektklasse, zu der das Objekt gehört. In der standardmäßigen Konfiguration für `inetOrgPerson` können beispielsweise bis zu sechs ACLs zu einem Benutzerobjekt hinzugefügt werden. Wenn eine LDAP-Suchanfrage ausgeführt wird, um dieses Benutzerobjekt mit allen Attributen zurückzugeben, dauert es etwas länger, dieses Objekt zum Client zurückzugeben, als das Benutzerobjekt ohne ACL-Attribute zurückzugeben.

Die standardmäßigen ACLs können deaktiviert werden. Dies ist jedoch von den Administratoren möglicherweise nicht erwünscht, da die ACLs für eine bessere Zugriffssteuerung erforderlich sind. Sie können jedoch die Suchleistung verbessern, indem Sie sie nicht anfordern oder als gefiltert gelesene Attribute kennzeichnen. Diese Änderungen führen nicht zum Anhalten von Anwendungen, weil die meisten Anwendungen effektive Berechtigungen verwenden und nicht auf bestimmten ACLs beruhen.

ACLs nicht anfordern: Ein ACL-Attribut wird von mehreren Anwendungen nicht benötigt. Diese Anwendungen können so geändert werden, dass sie bestimmte Attribute anfordern, die für die Anwendung relevant sind. Dies ermöglicht eine bessere Leistung beim Ausführen von LDAP-Suchen.

ACL als gefiltert gelesen kennzeichnen: Wenn eine Anwendung nicht geändert werden kann, kann das ACL-Attribut von einem Administrator über `arf_acl.ldif` als gefiltert gelesen gekennzeichnet werden. Wenn das ACL-Attribut als gefiltert gelesen gekennzeichnet wird, gibt der Server das Attribut nicht zurück, wenn für den Eintrag alle Attribute angefordert wurden. Wenn die LDAP-Suche jedoch ausgeführt wird, um operationelle Attribute zurückzugeben, oder wenn die Anforderung spezifisch nach ACL-Attributen fragt, wird das markierte Attribut zurückgegeben. Mit `rrf_acl.ldif` kann die Kennzeichnung „gefiltert gelesen“ eines ACL-Attributs deaktiviert werden. Die LDIFs beeinflussen das ACL-Attribut im Schema. Sie können daher nur von einem Benutzer mit Supervisor-Rechten im Baumstamm erweitert werden.

Standardmäßig ist ein ACL-Attribut nicht als gefiltert gelesen gekennzeichnet, sodass keine Leistungsverbesserung für Anforderungen zur Zurückgabe aller Attribute ersichtlich ist.

Die folgende Tabelle zeigt die Speicherorte von `arf_acl.ldif` und `rrf_acl.ldif` auf verschiedenen Plattformen.

Plattform	Standort
Linux	◆ `/opt/novell/eDirectory/lib64/nds-schema/`
Windows	◆ `<Speicherort der dekomprimierten Datei>\eDirectory\windows\x64\NDSonNT\ndsnt\nds`

Deaktivieren von ACL-Vorlagen

Sie können die Vorlagen der Zugriffssteuerungslisten (ACLs) deaktivieren, um die Sammelverarbeitungsleistung zu verbessern. Als Konsequenz fehlen daraufhin einige ACLs. Sie können dies jedoch beheben, indem Sie die erforderlichen ACLs zur LDIF-Datei hinzufügen oder später anwenden.

1. Führen Sie den folgenden Befehl aus:

   ```
   ldapsearch -D cn_of_admin -w password -b cn=schema -s base
   objectclasses=inetorgperson
   ```

   Auf diesen Befehl wird Folgendes zurückgegeben:

   ```
   dn: cn=schema
   ```

   ```
   objectClasses: (2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
   organizationalPerson STRUCTURAL MAY (groupMembership $ ndsHomeDirectory
   $ loginAllowedTimeMap $ loginDisabled $ loginExpirationTime $
   loginGraceLimit $ loginGraceRemaining $ loginIntruderAddress $
   loginIntruderAttempts $ loginIntruderResetTime $
   loginMaximumSimultaneous $ loginScript $ loginTime $
   networkAddressRestriction $ networkAddress $ passwordsUsed $
   passwordAllowChange $ passwordExpirationInterval $
   passwordExpirationTime $ passwordMinimumLength $ passwordRequired $
   passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile $
   publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
   minimumAccountBalance $ messageServer $ Language $ UID $
   lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
   higherPrivileges $ printerControl $ securityFlags $ profileMembership $
   Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
   sASSecretStoreData $ sASPKIStoreKeys $ userCertificate
   $ nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections
   $
   rADIUSAttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory
   $ rADIUSDefaultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
   $ rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
   carLicense $ departmentNumber $ employeeNumber $ employeeType $
   givenName $ homePhone $ homePostalAddress  $ initials $ jpegPhoto $
   labeledUri $ mail $ manager $ mobile $ pager $ ldapPhoto $
   preferredLanguage $ roomNumber $ secretary $ uid $ userSMIMECertificate
   $ x500UniqueIdentifier $ displayName $ userPKCS12) X-NDS_NAME 'User' X
   -NDS_NOT_CONTAINER '1' X-NDS_NONREMOVABLE '1' X-NDS_ACL_TEMPLATES
   ('2#subtree#[Self]#[All Attributes Rights]' '6#entry#[Self]#loginScript'
   '1#subtree#[Root Template]#[Entry Rights]'
   '2#entry#[Public]#messageServer' '2#entry#[Root Template]#groupMembership'
   '6#entry#[Self]#printJobConfiguration' '2#entry#[Root
   Template]#networkAddress'))
   ```

2. Löschen Sie die fett gedruckten Angaben aus der im vorigen Schritt notierten Ausgabe.
3. Speichern Sie die überarbeitete Ausgabe als LDIF-Datei.
4. Fügen Sie die folgenden Angaben zur neu gespeicherten LDIF-Datei hinzu:

   ```
   dn: cn=schema
   ```

   ```
   changetype: modify
   ```

```
delete: objectclasses

objectclasses: (2.16.840.1.113730.3.2.2)

-

add:objectclasses
```

Die LDIF-Datei ähnelt nun dem folgenden Beispiel:

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: (2.16.840.1.113730.3.2.2)

-

add:objectclasses

objectClasses: (2.16.840.1.113730.3.2.2 NAME 'inetOrgPerson' SUP
organizationalPerson STRUCTURAL MAY (groupMembership $ ndsHomeDirectory
$ loginAllowedTimeMap $ loginDisabled $ loginExpirationTime $
loginGraceLimit $ loginGraceRemaining $ loginIntruderAddress $
loginIntruderAttempts $ loginIntruderResetTime $
loginMaximumSimultaneous $ loginScript $ loginTime $
networkAddressRestriction $ networkAddress $ passwordsUsed $
passwordAllowChange $ passwordExpirationInterval $
passwordExpirationTime $ passwordMinimumLength $ passwordRequired
$ passwordUniqueRequired $ printJobConfiguration $ privateKey $ Profile
$
publicKey $ securityEquals $ accountBalance $ allowUnlimitedCredit $
minimumAccountBalance $ messageServer $ Language $ UID $
lockedByIntruder $ serverHolds $ lastLoginTime $ typeCreatorMap $
higherPrivileges $ printerControl $ securityFlags $ profileMembership $
Timezone $ sASServiceDN $ sASSecretStore $ sASSecretStoreKey $
sASSecretStoreData $ sASPKIStoreKeys $ userCertificate $
nDSPKIUserCertificateInfo $ nDSPKIKeystore $ rADIUSActiveConnections $
rADIUSAttributeLists $ rADIUSConcurrentLimit $ rADIUSConnectionHistory
$
rADIUSDefaultProfile $ rADIUSDialAccessGroup $ rADIUSEnableDialAccess
$ rADIUSPassword $ rADIUSServiceList $ audio $ businessCategory $
carLicense $ departmentNumber $ employeeNumber $ employeeType $
givenName $ homePhone $ homePostalAddress  $ initials $ jpegPhoto $
labeledUri $ mail
$ manager $ mobile $ pager $ ldapPhoto $ preferredLanguage $ roomNumber
$ secretary $ uid $ userSMIMECertificate $ x500UniqueIdentifier $
displayName $ userPKCS12) X-NDS_NAME 'User' X-ND S_NOT_CONTAINER '1' X
-NDS_NONREMOVABLE '1')
```

5 Geben Sie den folgenden Befehl ein:

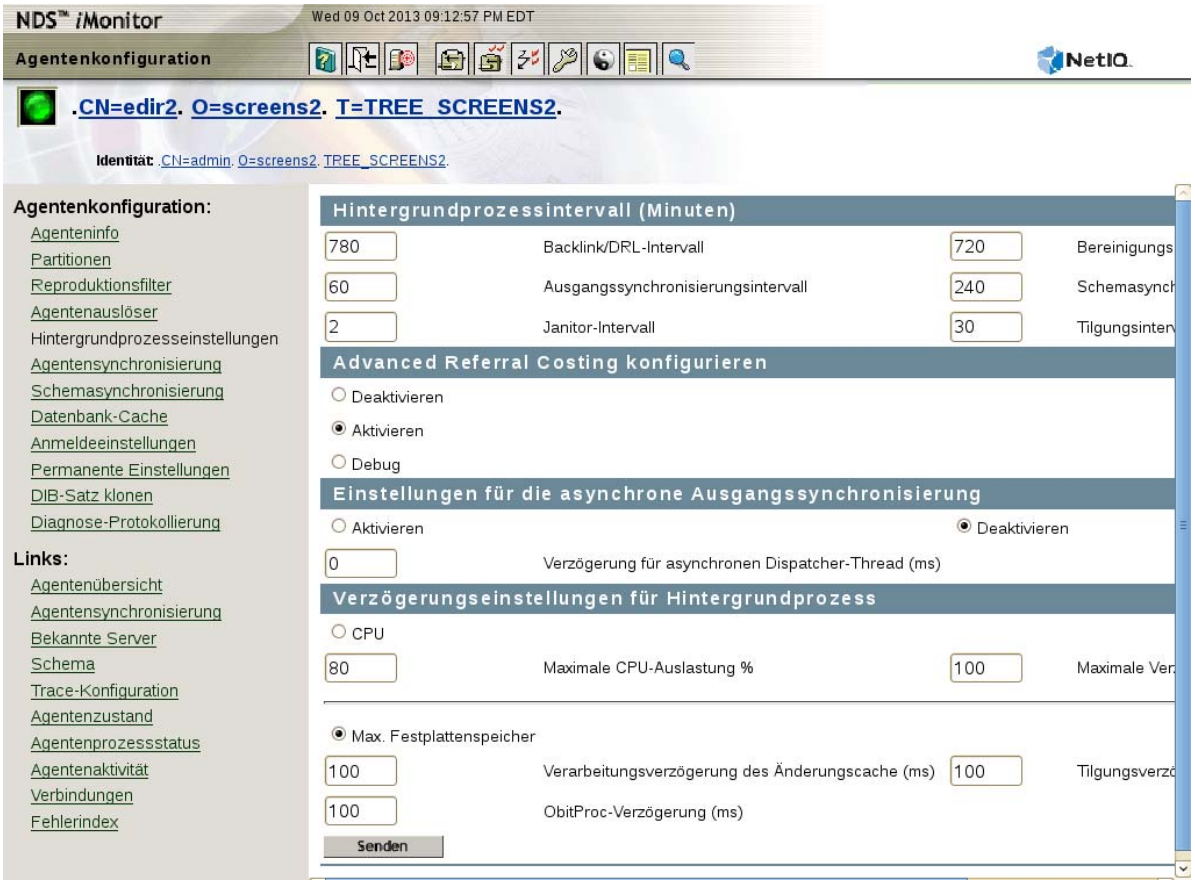
```
ldapmodify -D cn_of_admin -w password -f LDIF_file_name
```

Reproduktion

In dieser Version wurden bestimmte Hintergrundprozesse neu gestaltet, um besser für große, dynamische Umgebungen geeignet zu sein. Weitere Informationen finden Sie in „Managing Background Process“ (Verwalten von Hintergrundprozessen) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3)-Administrationshandbuch).

Wir empfehlen, das Hardlimit auf *5 ms* festzulegen und die asynchrone Ausgangssynchronisierung zu *aktivieren*. Wenn die Prozessauslastung jedoch steigt, erhöhen Sie die Inaktivitätsdauer. Abbildung 4-1 zeigt die Werte, die als **Verzögerungseinstellungen für Hintergrundprozess** festgelegt wurden.

Abbildung 4-1 *Hintergrundprozesseinstellungen*



Interne Labortests wurden mit einer Einrichtung aus 10 Servern und den folgenden Einstellungen ausgeführt: Hardlimit- *0 ms*, asynchrone Ausgangssynchronisierung - *aktiviert* Verzögerung für asynchronen Dispatcher-Thread - *0 ms*. Die Tests haben ergeben, dass die Reproduktion 7 Mal schneller ist als mit den standardmäßigen Einstellungen. Während des Tests wurden keine weiteren Clientvorgänge ausgeführt.

HINWEIS: Um aus diesen Skalierbarkeitsverbesserungen die besten Ergebnisse für Ihre Systeme zu erzielen, muss auf allen Servern OpenText eDirectory 9.1 oder höher ausgeführt werden. Auch wenn im Reproduktionsring einige ältere Versionen vorhanden sind, wird eine Leistungsverbesserung erreicht.

Solid State Disk (SSD)

Diese Version unterstützt Enterprise SSD zur Verbesserung der E/A-Operationen. Tabelle 4-1 auf Seite 26 zeigt die Verbesserungen bei der SSD-Reparatur in der Testeinrichtung:

Tabelle 4-1 *Reparaturleistung*

DIB-Größe (GB)	HDD (Dauer in Minuten)	SSD (Dauer in Minuten)	% Verbesserung
11	80	53	33,75
24	277	169	38,98
34	542	296	45,38
75	1383	618	55,31
98	3171	1023	67,73

NMAS-Anmeldeaktualisierungsintervall

Weitere Informationen finden Sie in „Using the sasUpdateLoginInfo and sasUpdateLoginTimeInterval Attribute“ (Arbeiten mit den Attributen „sasUpdateLoginInfo“ und „sasUpdateLoginTimeInterval“) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3)-Administrationshandbuch).

SSL-Overhead

Aufgrund der Verschlüsselungsanforderungen fügt LDAP über SSL eine zusätzliche Last für den Prozessor hinzu. Laboruntersuchungen zeigen eine Auswirkung von mehr als 10 % auf die Leistung durch den Verschlüsselungs-Overhead.

Import/Konvertierung/Export (Import Convert Export, ICE)

Das OpenText Import Convert and Export (ICE)-Dienstprogramm arbeitet mit einem optimierten Sammelaktualisierungsprotokoll, LBURP, um Daten zu OpenText eDirectory heraufzuladen. Mit diesem Protokoll können Daten deutlich schneller heraufgeladen werden als mit dem einfachen Befehl `ldapmodify`. Weitere Informationen finden Sie unter Offline Bulkload Utility (Offline-Massenladeprogramm) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide (OpenText™ eDirectory CE 24.4 (v9.3)-Administrationshandbuch).*

ldif2dib

Informationen zur Optimierung der OpenText eDirectory-Leistung beim Offline-Bulkload mit dem ldif2dib-Dienstprogramm finden Sie in Tuning ldif2dib (Optimierung von ldif2dib) im *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (OpenText™ eDirectory CE 24.4 (v9.3)-Administrationshandbuch).

Größere NCP-Pakete

Zur Kommunikation mit unterschiedlichen Servern verwendet OpenText eDirectory NCP (NetWare Core Protocol, NetWare-Kernprotokoll). In früheren Releases unterstützte NCP eine Paketgröße von maximal 64 KB, was den Höchstdurchsatz bei der Datenübertragung per NCP deutlich einschränkte. In dieser Version unterstützt NCP Paketgrößen bis 1 MB, sodass OpenText eDirectory bis zu 1 MB an Daten in nur einem Paket synchronisieren kann. OpenText eDirectory beginnt die Synchronisierung bei einer Paketgröße von 64 KB und erhöht diese dann je nach der verbleibenden Menge an zu synchronisierenden Daten. Dies verbessert die Reproduktionsleistung in erheblichem Maße. Wenn Ihre beiden Server unter Version 9.2 ausgeführt werden, ist zur Nutzung dieser Verbesserung keine zusätzliche Konfiguration erforderlich.

5 OpenText eDirectory-Konfiguration

Dieser Abschnitt enthält folgende Informationen:

- „Konfigurieren des FLAIM-Subsystems“, auf Seite 29
- „Ändern der FLAIM-Cache-Einstellungen“, auf Seite 29

Konfigurieren des FLAIM-Subsystems

OpenText eDirectory bietet zwei Mechanismen zur Steuerung des Cache-Speicherverbrauchs, um den verschiedensten Bereitstellungen und Konfigurationen gerecht zu werden. Die Mechanismen schließen sich gegenseitig aus.

- „Cache-Hardlimit“, auf Seite 29
- „Dynamische Anpassung der Grenze“, auf Seite 29

Cache-Hardlimit

Es gibt die folgenden Möglichkeiten, um einen festen Arbeitsspeichergrenzwert festzulegen:

- Als fester Bytewert.
- Als Prozentsatz des physischen Arbeitsspeichers.
- Als Prozentsatz des verfügbaren physischen Arbeitsspeichers.

Wenn ein Hardlimit mit der zweiten oder dritten Methode festgelegt wird, wird dies immer in einen festen Bytewert umgerechnet. Dies bedeutet, dass die Bytezahl für die zweite Methode einem Prozentsatz des beim Starten von OpenText eDirectory erkannten physischen Arbeitsspeichers entspricht. Bei der dritten Methode entspricht die Bytezahl einem Prozentsatz des beim Starten von OpenText eDirectory erkannten, verfügbaren physischen Arbeitsspeichers.

Dynamische Anpassung der Grenze

Bei der dynamischen Anpassung passt OpenText eDirectory als Reaktion auf den variablen Speicherverbrauch anderer Prozesse regelmäßig den eigenen Speicherverbrauch an. In üblichen Szenarien funktioniert dieser Mechanismus gut. Er wird jedoch für eine optimale Leistung von OpenText eDirectory auf Linux-Plattformen nicht empfohlen, weil die Speicherverwendungsmuster und Speicherzuweisungen auf Linux-Plattformen starke Unterschiede aufweisen.

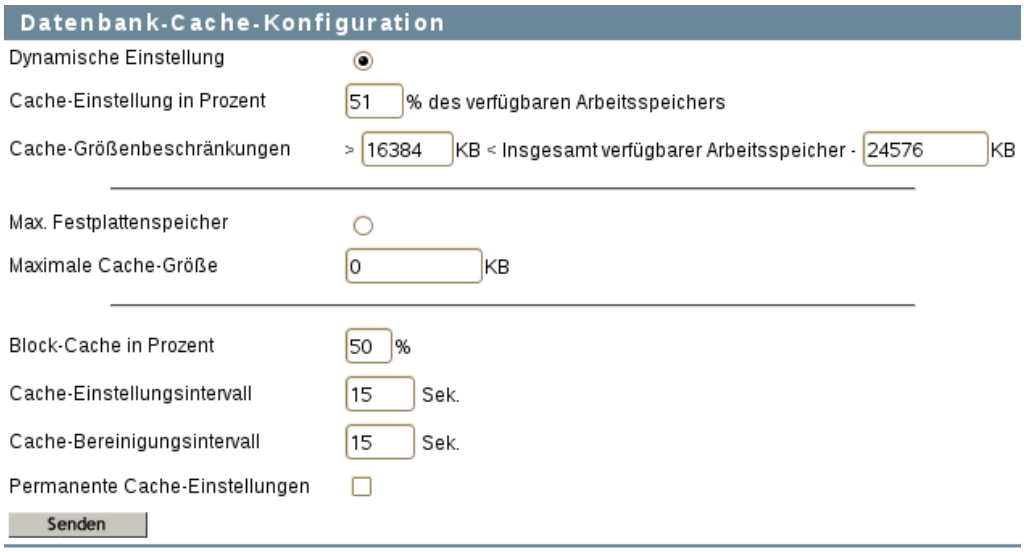
Ändern der FLAIM-Cache-Einstellungen

- „Ändern der FLAIM-Cache-Einstellungen über OpenText iMonitor“, auf Seite 30
- „Ändern der FLAIM-Cache-Einstellungen über _ndsdb.ini“, auf Seite 31

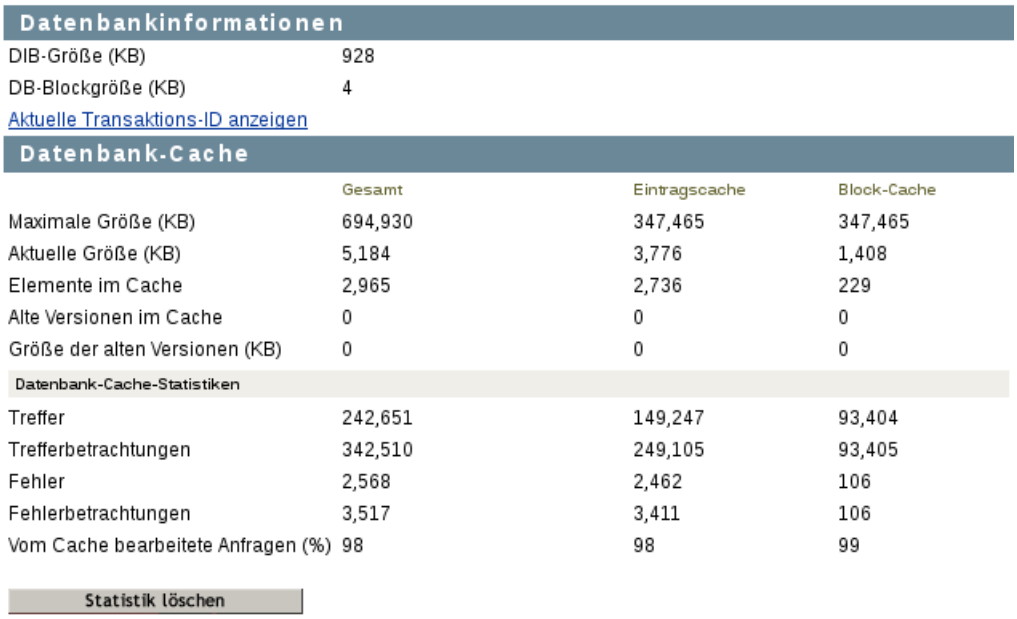
Ändern der FLAIM-Cache-Einstellungen über OpenText iMonitor

Sie können OpenText iMonitor für folgende Aufgaben verwenden:

- Cache-Einstellungen anzeigen oder ändern



- Cache-Statistiken überwachen



Siehe Datenbank-Cache unter der Agentenkonfiguration von OpenText iMonitor für die obigen Informationen.

Informationen zum Datenbank-Cache	Beschreibung
Maximale Größe	Die maximale Größe (in KB), auf die der angegebene Cache anwachsen darf.
Aktuelle Größe	Die aktuelle Größe (in KB) des angegebenen Cache.
Elemente im Cache	Die Anzahl der Elemente im angegebenen Cache.
Alte Versionen im Cache	Die Anzahl der alten Versionen im angegebenen Cache. Alte Versionen von Cache-Elementen werden beibehalten, um die Konsistenz der Lesetransaktionen in der Datenbank aufrechtzuerhalten. In anderen Worten ausgedrückt: Wenn sich ein Thread in einer Lesetranskation und ein anderer in einer Schreibtransaktion befindet, werden die alten Versionen der vom Schreiber geänderten Blöcke im Namen des Lesers aufrechterhalten. Dies gewährleistet, dass die Ergebnisse des Lesers während der gesamten Dauer der Transaktion eine konsistente Anzeige produzieren, auch wenn währenddessen Änderungen vorgenommen wurden.
Größe der alten Versionen	Die Größe (in KB) der alten Versionen im Cache.
Treffer	Die Anzahl der Male, bei denen im angegebenen Cache erfolgreich auf das Element zugegriffen wurde.
Trefferbetrachtungen	Die Anzahl der Elemente im Cache, die betrachtet wurden, bevor auf das Element im angegebenen Cache erfolgreich zugegriffen wurde. Das Verhältnis von Treffern zu Trefferbetrachtungen ist eine Angabe der Cache-Sucheffizienz. Im Normalfall sollte das Verhältnis nahe 1:1 liegen.
Fehler	Die Anzahl der Male, bei denen ein Element nicht im angegebenen Cache gefunden wurde und aus einem Cache niedrigerer Ebene oder von der Festplatte abgerufen werden musste.
Fehlerbetrachtungen	Die Anzahl der Elemente, die im Cache betrachtet wurden, bevor bestimmt wurde, dass das gewünschte Element im angegebenen Speicher nicht vorhanden ist. Das Verhältnis von Fehlern zu Fehlerbetrachtungen ist eine Angabe der Cache-Sucheffizienz. Im Normalfall sollte das Verhältnis nahe 1:1 liegen.

Ändern der FLAIM-Cache-Einstellungen über _ndsdb.ini

Die FLAIM-Cache-Einstellungen und andere FLAIM-Konfigurationen können durch Ändern der Datei `_ndsdb.ini` im DIB-Verzeichnis vorgenommen werden. Starten Sie OpenText eDirectory neu, wenn die Datei `_ndsdb.ini` geändert wurde.

Sie können eine dynamisch angepasste oder eine feste Grenze für den Cache festlegen. Die Cache-Optionen sind unten aufgeführt. Mehrere Optionen können in beliebiger Reihenfolge, getrennt durch Kommas, angegeben werden. Alle sind optional.

- **DYN oder HARD** - Dynamisches Anpassen der Grenze oder Festlegen eines Hardlimit.

- **% : Prozentsatz** - Prozentsatz an verfügbarem oder physischem Arbeitsspeicher, der verwendet werden soll.
- **AVAIL oder TOTAL** - Legt fest, ob sich der Prozentsatz auf den verfügbaren oder gesamten physischen Arbeitsspeicher bezieht. Der Parameter gilt nur für das Hardlimit und wird für dynamisch angepasste Grenzen ignoriert. Dynamisch angepasste Grenzen werden stets auf Grundlage des verfügbaren physischen Arbeitsspeichers berechnet. Standardmäßig ist AVAIL festgelegt.
- **MIN: Byte** - Mindestanzahl von Byte.
- **MAX: Byte** - Höchstanzahl von Byte.
- **LEAVE: Byte** - Mindestanzahl von Byte, die übrig sein sollen.

  Beispiel:

  ```
  cache=HARD,%:75, MIN:200000000
  ```

  ```
  cache=500000000
  ```

- **preallocatecache: wahr/falsch** - Diese Einstellung führt dazu, dass OpenText eDirectory die durch das Cache-Hardlimit angegebene Menge Speicher vorab zuweist.
- **rfldirectory -** Für RFL-Dateien kann ein anderer Pfad angegeben werden.
- **cpinterval** - Dauer in Sekunden, nachdem FLAIM einen Checkpoint erzwingt. Der Standardwert ist 3 Minuten.
- **maxdirtycache** - Höchstwert in Byte für den „dirty“ Cache.
- **lowdirtycache** - Mindestwert in Byte für den „dirty“ Cache.
- **blockcachepercent** - Prozentsatz des FLAIM-Cache, der für den Block-Cache verwendet wird.
- **cacheadjustinterval** - Intervall in Sekunden zur dynamischen Anpassung des Cache.
- **cachecleanupinterval** - Intervall in Sekunden zum Löschen älterer Versionen von Einträgen und Blöcken aus dem Cache.