opentext™

OpenText™ eDirectory CE 24.4 (v9.3)

Guida all'ottimizzazione

Gennaio 2025

Note legali

Copyright 2009-2024 Open Text Tutti i diritti riservati.

Le uniche garanzie per i prodotti e servizi di Open Text e dei suoi affiliati e licenziatari ("Open Text") sono quelle indicate nelle dichiarazioni di garanzia esplicite che accompagnano tali prodotti e servizi. Nulla di quanto riportato nel presente documento deve essere interpretato come garanzia aggiuntiva. Open Text non sarà da ritenersi responsabile per errori tecnici o editoriali contenuti nel presente documento né per eventuali omissioni. Le informazioni di questo documento sono soggette a modifiche senza preavviso.

Sommario

Informazioni sulla guida e sulla libreria 5

1 Panoramica 7

Prerequisiti . . . 7

2 Sottosistemi di OpenText eDirectory 9

Database FLAIM . . . 9
- Punto di controllo . . . 10
- Indici . . . 10
- Log di roll forward . . . 11
- Containerizzazione degli attributi FLAIM . . . 11

Pool di thread . . . 12

3 Analisi dei colli di bottiglia di sistema 15

Sottosistema I/O del disco . . . 15
Sottosistema CPU . . . 16
Sottosistema di memoria . . . 16
Sottosistema di rete . . . 17

4 Ottimizzazione dei sottosistemi OpenText eDirectory 19

Database FLAIM . . . 19
- Scelta degli indici . . . 20
- Ottimizzazione degli aggiornamenti . . . 20

Pool di thread . . . 21
ACL . . . 21
- Miglioramento delle letture e delle ricerche di OpenText eDirectory . . . 21
- Disabilitazione dei modelli ACL . . . 22

Replica . . . 24
Disco allo stato solido (SSD) . . . 25
Intervallo di aggiornamento di login NMAS . . . 26
Overhead SSL . . . 26
Import Convert and Export (ICE) . . . 26
ldif2dib . . . 26
Dimensione del pacchetto NCP migliorata . . . 26

5 Configurazione di OpenText eDirectory 29

Configurazione del sottosistema FLAIM . . . 29
- Limite rigido della cache . . . 29
- Regolazione dinamica del limite . . . 29

Modifica delle impostazioni della cache FLAIM . . . 29
- Modifica delle impostazioni della cache FLAIM mediante OpenText iMonitor . . . 30

Modifica delle impostazioni della cache FLAIM mediante _ndsdb.ini 32

Informazioni sulla guida e sulla libreria

In questa guida viene descritto come analizzare e ottimizzare OpenText™ eDirectory CE 24.4 (v9.3) per ottenere prestazioni elevate in tutte le installazioni.

Per la versione più recente della *Guida all'ottimizzazione di OpenText™ eDirectory*, consultare il sito Web della documentazione online di OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/).

Destinatari

Questa guida è destinata agli amministratori di rete.

Altre informazioni incluse nella libreria

La libreria contiene le risorse seguenti:

Guida all'amministrazione di XDASv2

Descrive come configurare e utilizzare XDASv2 per il controllo di OpenText eDirectory e OpenText Identity Manager.

Guida all'installazione

Descrive come installare OpenText eDirectory ed è destinata agli amministratori di rete.

Guida all'amministrazione

Descrive come gestire e configurare OpenText eDirectory.

Guida alla risoluzione dei problemi

Descrive come risolvere i problemi relativi a OpenText eDirectory.

Guida alle novità

Descrive le nuove funzionalità di OpenText eDirectory.

Le guide sono disponibili sul sito Web della documentazione di OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/) .

Per informazioni sull'utility di gestione di eDirectory, consultare la Guida all'amministrazione di OpenText™ Identity Console (https://www.netiq.com/documentation/identity-console/).

1 Panoramica

OpenText eDirectory 9.3 è una soluzione di servizi di directory ad alte prestazioni, conforme agli standard, multipiattaforma, altamente scalabile e a tolleranza di errore. In questa guida vengono fornite informazioni relative all'ottimizzazione dell'ambiente OpenText eDirectory che consentono di potenziarne le prestazioni.

L'ottimizzazione delle prestazioni è un'attività complessa. Per realizzare tale attività, è necessario comprendere i sottosistemi sia di OpenText eDirectory sia del sistema operativo. L'ottimizzazione implica il monitoraggio del sistema allo scopo di identificare i colli di bottiglia e risolverli uno alla volta. Spesso, le risorse risultano ridotte e l'ottimizzazione si limita solo a OpenText eDirectory e al sistema operativo.

Nella Guida, leggere la sezione Prerequisiti prima di tentare qualsiasi tipo di ottimizzazione, quindi, passare alle altre sezioni. Nel capitolo Sottosistemi di OpenText eDirectory vengono descritti i sottosistemi principali che interessano le prestazioni di OpenText eDirectory. Nel capitolo Analisi dei colli di bottiglia di sistema vengono descritte le varie risorse di sistema e la relativa influenza sulle prestazioni OpenText eDirectory. Nel capitolo Ottimizzazione dei sottosistemi OpenText eDirectory viene descritto come analizzare e ottimizzare OpenText eDirectory in varie condizioni e ambienti. Infine, nel capitolo Configurazione di OpenText eDirectory viene descritto come configurare vari parametri che possono essere ottimizzati.

Prerequisiti

Prima di tentare l'ottimizzazione del sistema delle prestazioni, assicurarsi che i seguenti prerequisiti generali siano soddisfatti:

- Un'efficace progettazione della struttura ad albero di OpenText eDirectory consente di migliorarne le prestazioni. Infatti, possono essere applicate le seguenti considerazioni:
  - Le applicazioni leggono tutte le informazioni localmente sul server senza dover concatenare le richieste.
  - OpenText eDirectory gestisce automaticamente in modo efficiente i riferimenti agli oggetti. Se possibile, gli oggetti residenti in un server non devono fare riferimento a oggetti che non sono locali su tale server, in quanto mantenere riferimenti a oggetti non locali può richiedere più tempo. Se tali riferimenti esistono, è necessario mantenere i backlink. Ciò diventa più complesso nelle installazioni di grandi dimensioni.
  - Se è necessario un gruppo con 10.000 membri o più, è consigliato utilizzare i gruppi dinamici. In questo modo, infatti, è possibile evitare l'overhead associato alla conservazione dei riferimenti per molte persone. Scegliere la configurazione del gruppo dinamico con estrema attenzione, in quanto l'utilizzo di più gruppi dinamici con criteri di ricerca non corretti potrebbe sovraccaricare il server e ridurne le prestazioni complessive. Se il completamento di un'operazione di ricerca richiede molto tempo, l'indice selezionato potrebbe risultare inefficace. Ridurre al minimo l'utilizzo dei gruppi regolari (statici) in quanto può incrementare la ricerca nella struttura ad albero durante il login.

- Utilizzare gli ACL in modo efficiente. Utilizzare, ad esempio, il trustee e assegnarlo a livello di container anziché utilizzare il modello ACL che assegna diritti a se stesso. Minore è il numero di ACL, migliori sono le prestazioni. Per ulteriori informazioni sugli ACL, vedere "OpenText™ eDirectory Rights" (Diritti di OpenText™ eDirectory) nella *OpenText eDirectory CE 24.4 (v9.3) eDirectory Administration Guide* (Guida all'amministrazione di eDirectory di OpenText™ eDirectory CE 24.4 (v9.3)).
- Distribuire il carico su più server di replica.
- Sebbene un'efficace struttura ad albero consenta di ridurre al minimo la necessità di eseguire ricerche nella struttura ad albero, talvolta risulta comunque necessario. È possibile prendere in considerazione l'"Advanced Referral Costing" (Determinazioni costi riferimenti avanzata) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guida all'amministrazione di eDirectory di OpenText™ eDirectory CE 24.4(v9.3)).
- Se l'esecuzione dei login è lenta, è possibile disabilitarne gli aggiornamenti. Esistono diversi modi per disabilitare gli aggiornamenti del login per NDS e per il servizio NetIQ NMAS (Modular Authentication Service, servizio di autenticazione modulare). Tuttavia, è importante comprendere le implicazioni di sicurezza (http://www.novell.com/documentation/nmas33/admin/data/bg8dphs.html).

- Eseguire controlli dello stato tramite iMonitor. Per ulteriori informazioni, vedere "Viewing OpenText Server Health" (Visualizzazione dello stato del server OpenText eDirectory) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guida all'amministrazione di OpenText™ eDirectory CE 24.4 (v9.3)). Verificare quanto segue:
  - L'orario è sincronizzato in tutti i server di replica.
  - I processi di sincronizzazione delle repliche e i processi in background sono integri.

2 Sottosistemi di OpenText eDirectory

In questa sezione vengono illustrati i sottosistemi di OpenText eDirectory.

- “Database FLAIM” a pagina 9
- “Pool di thread” a pagina 12

Database FLAIM

OpenText eDirectory utilizza FLAIM come database. FLAIM (Flessibile Adattabile Information Manager) viene utilizzato per le informazioni tradizionali, volatili e complesse. È un motore di database molto scalabile che supporta più lettori e un modello di concorrenza a scrittura singola. I lettori non bloccano gli scrittori e questi non bloccano i lettori.

Fisicamente, FLAIM organizza i dati in blocchi. Generalmente, alcuni blocchi vengono conservati in memoria e costituiscono la cache di blocchi. La cache voci (detta anche cache di record) memorizza nella cache le voci logiche provenienti dal database. Le voci vengono costruite da elementi nella cache di blocchi. FLAIM conserva le tabelle hash per entrambe le cache. La dimensione del compartimento di hash viene periodicamente regolata in base al numero di elementi.

Di default, OpenText eDirectory utilizza una dimensione del blocco di 4 KB. La dimensione della cache di blocchi per la memorizzazione nella cache dell'intero DIB è uguale alla dimensione del DIB mentre la dimensione richiesta per la cache voci è pari a circa due/quattro volte la dimensione del DIB.

Durante il recupero di una voce, FLAIM verifica innanzitutto la voce nella cache voci. Se la voce esiste, la lettura dalla cache di blocchi non è necessaria. Durante il recupero di un blocco dal disco, FLAIM verifica innanzitutto il blocco nella cache. Se non esiste, la lettura del disco non è necessaria.

Quando una voce viene aggiunta o modificata, il commit dei blocchi corrispondenti per tale voce non viene eseguito direttamente sul disco. Perciò, il disco e la memoria potrebbero non essere sincronizzati. Tuttavia, gli aggiornamenti realizzati per la voce vengono registrati nell'RFL (Roll-Forward Log, log di roll forward). Un RFL viene utilizzato per ripristinare le transazioni dopo un errore di sistema.

LRU (Least Recently Used, utilizzati meno di recente) è l'algoritmo di sostituzione utilizzato per la sostituzione degli elementi nella cache.

- “Punto di controllo” a pagina 10
- “Indici” a pagina 10
- “Log di roll forward” a pagina 11
- “Containerizzazione degli attributi FLAIM” a pagina 11

Punto di controllo

Un punto di controllo porta la versione su disco del database allo stesso stato coerente del database presente nella memoria (cache). FLAIM è in grado di eseguire un punto di controllo durante l'attività di aggiornamento minimo sul database. Viene eseguito ogni secondo e scrive i blocchi modificati (cache modificata) sul disco. I blocchi che vengono modificati nella cache, ma non ancora scritti sul disco sono denominati "blocchi modificati". FLAIM acquisisce un blocco del database ed esegue la quantità massima di lavoro possibile fino al completamento del punto di controllo o fino a quando non vi sia un altro thread in attesa per l'aggiornamento del database. Per evitare una sincronizzazione non sufficiente del database, sussistono alcune condizioni in cui un punto di controllo viene imposto anche se i thread sono in attesa per l'aggiornamento del database:

- Se il thread del punto di controllo non riesce a completare un punto di controllo all'interno di un intervallo di tempo specificato (il valore di default è 3 minuti), viene forzato e la cache modificata viene pulita.
- Se la dimensione della cache modificata è maggiore della `maxdirtycache` (se impostata), viene forzato un punto di arresto per ridurre le dimensioni della cache modificata riducendola a quelle della `mindirtycache` (se impostata) o a zero.

Indici

Un indice è un set di chiavi disposte in modo tale da velocizzare notevolmente l'attività di ricerca di una particolare chiave all'interno dell'indice. Le chiavi dell'indice vengono costruite estraendo i contenuti di uno o più campi (attributi) dalle voci. Gli indici sono mantenuti nella cache di blocchi. Qualsiasi modifica agli attributi indicizzati richiede modifiche nei blocchi dell'indice.

OpenText eDirectory definisce un set di indici di default per gli attributi di sistema (campi). Attributi di sistema come `parentID` e `ancestorID` vengono utilizzati per le ricerche a livello unico e per quelle elaborate nel sottoalbero. Questi indici non possono essere sospesi o eliminati. La directory li utilizza internamente. Gli indici di default sono definiti per attributi quali CN, `cognome`, `nome specificato` e così via. Gli indici possono essere di tipo presenza, valore e stringa secondaria. Tali indici possono essere sospesi. Se eliminati, vengono ricreati automaticamente.

Per creare gli indici, è possibile utilizzare OpenText Identity Console o l'utility ndsindex Lightweight Directory Access Protocol (LDAP). Gli Indici (http://www.netiq.com/documentation/edir83/) sono specifici del server.

Abilitando il tag di Gestione archivio (`StrMan`) in DSTrace (ndstrace), è possibile visualizzare l'indice scelto per le interrogazioni di ricerca.

L'esempio riportato di seguito è per un log DSTrace, in cui si prevede una ricerca del sottoalbero utilizzando `"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
```

L'esempio riportato di seguito è per un log DSTrace, in cui si prevede una ricerca del sottoalbero utilizzando`"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
```

Per migliorare le prestazioni di ricerca con l'ordinamento sul lato server, utilizzare l'opzione -a per aggiungere come prefisso l'attributo `AncestorID` all'elenco di attributi trasferiti quando si crea un nuovo indice.

Nella tabella seguente viene illustrata la tendenza all'aumento delle dimensioni di DIB durante la creazione di indici semplici/composti per un determinato numero di oggetti. Si consiglia di fare riferimento ai dati di prova durante la creazione di indici semplici/composti:

Numero di oggetti	Aumento della dimensione di DIB con indice semplice contenente una dimensione dell'attributo di 20 caratteri	Aumento della dimensione di DIB con indice semplice contenente una dimensione dell'attributo di 200 caratteri	Aumento delle dimensioni di DIB con un indice composto contenente una dimensione dell'attributo di 200 caratteri
23 K	24 KB	3 MB	7 MB
47 K	50 KB	6 MB	15 MB
0.5 milioni	4 MB	55 MB	100 MB
1 milioni	9 MB	111 MB	210 MB

Log di roll forward

FLAIM registra le operazioni per ogni transazione di aggiornamento in un file dell'RFL. Un RFL viene utilizzato per recuperare le transazioni da un errore di sistema o durante il ripristino da un backup. Il file RFL viene troncato dopo il completamento di ogni punto di controllo a meno che non sia attivato (`rflkeepfiles`) utilizzando un backup costante a caldo (http://www.netiq.com/documentation/edir93/).

Containerizzazione degli attributi FLAIM

Per garantire un utilizzo ottimale della cache voci e prestazioni migliorate delle operazioni di ricerca degli attributi, FLAIM memorizza gli attributi con i valori più grandi o con il numero di valori più elevato in un'ubicazione separata, ovvero, nel container di attributi. Lo spostamento di default nel container viene eseguito automaticamente quando l'attributo:

- ha più di 25 valori
- ha un valore superiore a 2048 byte

Per disabilitare la containerizzazione automatica degli attributi, aggiungere `disablemovetoattrcontainer =1` nel file `_ndsdb.ini`, quindi riavviare OpenText eDirectory.

Grazie alla flessibilità fornita da OpenText eDirectory, adesso è possibile pianificare lo spostamento degli attributi. Prima, vengono visualizzati gli attributi che sono pronti per essere spostati, quindi, è possibile pianificare lo spostamento in un secondo momento, in base alle proprie esigenze.

Per visualizzare il numero di attributi pronti per essere spostati ai container di attributi, eseguire il comando `ndscheck`. Per visualizzare i dettagli degli attributi, utilizzare l'attributo **dsContainerReadyAttrs** di iMonitor negli oggetti Pseudo server in **Configurazione dell'agente**. L'utente può trovare anche gli attributi contrassegnati per l'indicizzazione in Stato dell'agente.

È possibile avviare la containerizzazione degli attributi utilizzando l'opzione Riparazione singolo oggetto del comando `ndsrepair` per l'oggetto Pseudo Server. Per eseguire la containerizzazione di un attributo, eseguire il comando `ndsrepair` con il nuovo switch `-am` avanzato seguito dal nome dell'attributo, come mostrato di seguito:

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

Dopo aver spostato un attributo al container di attributi, OpenText eDirectory crea un indice del sistema con il nome dell'attributo. Quando viene eseguita la containerizzazione di un attributo, non è possibile spostarlo nuovamente al container originale.

Nota: se un attributo ha un valore maggiore di 2048 byte, viene comunque spostato nel container ma OpenText eDirectory non crea alcun indice di sistema.

Pool di thread

OpenText eDirectory è multithreading per questioni relative alle prestazioni. Nel multithreading, quando il sistema è occupato, vengono creati ulteriori thread per gestire il carico e alcuni thread vengono interrotti per evitare overhead in eccesso. La creazione e la distruzione frequente di thread sono operazioni inefficienti e costose. Anziché generare nuovi thread e distruggerli per ogni task, un numero di thread viene avviato e posizionato in un pool. Il sistema alloca i thread del pool di thread a tanti task quanti ne sono richiesti. I task sono disposti in due tipi di code:

- I task che richiedono una pianificazione immediata risiedono nella coda Pronti.
- I task, invece, la cui pianificazione può essere realizzata in un secondo momento risiedono nella coda In attesa.

Non tutti i moduli utilizzano il pool di thread. Il numero effettivo di thread per il processo è superiore a quello presente nel pool di thread. Ad esempio, FLAIM gestisce separatamente i propri thread in background.

L'esecuzione del comando `ndstrace - c threads` restituisce le seguenti statistiche del pool di thread:

- Il numero totale di thread che vengono generati, interrotti e inattivi.
- Il numero totale di thread di lavoro attuali e il numero massimo di thread di lavoro.
- Il numero di task e il numero massimo di task presenti nella coda Pronti.
- Il numero minimo, massimo e medio di microsecondi trascorsi nella coda Pronti.
- Il numero attuale di task nella coda In attesa.

Un esempio di un campione di pool di thread:

```
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
```

Esistono determinati parametri del pool di thread:

- **n4u.server.max-threads:** Numero massimo di thread che possono essere disponibili nel pool.
- **n4u.server.idle-threads:** Numero massimo di thread inattivi che possono essere disponibili nel pool.
- **n4u.server.start-threads:** Numero di thread avviati.

Eseguire i comandi `ndsconfig get` e `ndsconfig set` per recuperare e impostare la dimensione del pool di thread.

3 Analisi dei colli di bottiglia di sistema

Esistono varie risorse di sistema che influenzano le prestazioni di OpenText eDirectory. Inoltre, l'aggiornamento alla versione più recente del sistema operativo migliora le prestazioni.

- “Sottosistema I/O del disco” a pagina 15
- “Sottosistema CPU” a pagina 16
- “Sottosistema di memoria” a pagina 16
- “Sottosistema di rete” a pagina 17

Sottosistema I/O del disco

Il sottosistema del disco rappresenta il collo di bottiglia più comune. Il modulo I/O richiede tempi relativamente più lunghi con code più estese, provocando in questo modo un utilizzo del disco elevato e cicli inattivi della CPU. Utilizzare lo strumento iostat durante i carichi di picco previsti per determinare gli indicatori del tempo medio di risposta.

Le operazioni di lettura, scrittura e aggiornamento su disco possono essere sequenziali o casuali. Gli aggiornamenti e le letture casuali rappresentano il modello di accesso più comune nelle installazioni di OpenText eDirectory.

Alcune soluzioni per workload casuali:

- Aumentare la RAM. Ciò consente di memorizzare nella cache i dati utilizzati frequentemente o i dati read-ahead a livello di file system. Consente inoltre di memorizzare nella cache il DIB all'interno del sottosistema FLAIM.
- Utilizzare volumi dedicati per il DIB. Le prestazioni del file system migliorano nei volumi creati in una posizione più vicina all'asse. Utilizzare volumi dedicati per l'RFL e altri log.
- In quanto a causa della frammentazione i dischi sviluppano un aumento di latenza in un determinato periodo di tempo, è necessario deframmentarli.
- Aggiungere unità disco separate per l'RFL FLAIM. Questo tipo di registrazione può essere eseguita su dischi ad alta velocità.
- Utilizzare un ambiente RAID 10 (1+0) con più unità disco.

I file creati da OpenText eDirectory possono aumentare fino a 4 GB. I file system ottimizzati per gestire file di grandi dimensioni funzionano in modo efficiente con OpenText eDirectory.

- Per Solaris™, il file system Veritas* VxFS è un file system basato sull'estensione dove i metadati del file system sono ottimizzati per file di grandi dimensioni. Il file system UFS è indirettamente basato su blocchi in cui i metadati del file system vengono memorizzati in un numero maggiore di blocchi. UFS può anche essere distribuito per file di grandi dimensioni, ma, in tal caso, il rendimento risulta più lento.
- Per Linux™, il file system Reiser è un file system con journaling veloce e offre prestazioni migliori rispetto al file system ext3 su grandi set DIB. Tuttavia, è noto che la modalità di write-back del journaling di ext3 corrisponde alle prestazioni del file system Reiser, sebbene la modalità

ordinata di default fornisca una maggiore coerenza dei dati. XFS è un file system con journaling ad alte prestazioni, in grado di gestire file di grandi dimensioni e offrire trasferimenti di dati senza alcun problema. OpenText eDirectory 9.3 è supportato su piattaforme SLES 15 a 32 e 64 bit con file system XFS.

FLAIM supporta una dimensione di blocco di 4 KB e 8 KB. Il valore di default è 4 KB. È lo stesso valore della dimensione di blocco di default su Linux (`tune2fs - l dispositivo`). Tuttavia, su Solaris, il file system UFS viene creato con una dimensione di blocco di default pari a 8 KB (`df -g punto di montaggio`). Se la dimensione di blocco FLAIM è inferiore alla dimensione di blocco del file system, possono verificarsi scritture parziali del blocco. Se la dimensione di blocco del database è superiore alla dimensione di blocco del file system, le letture e le scritture singole del blocco vengono suddivise in una serie di operazioni di I/O fisiche distinte. Pertanto, è consigliabile mantenere sempre la dimensione di blocco FLAIM uguale a quella del file system.

Le dimensioni di blocco possono essere controllate solo durante la creazione del DIB. Aggiungere una riga "blocksize=8192" `_ndsdb.ini` per creare il DIB con dimensione di blocco di 8 KB.

La scelta della dimensione di blocco esatta dipende dalla dimensione media del record FLAIM presente nelle installazioni dell'utente. Sono richieste prove empiriche elaborate sul set appropriato dei dati della prova onde determinare la dimensione di blocco adatta per una specifica installazione.

Sottosistema CPU

OpenText eDirectory si basa su un'architettura altamente scalabile. Le prestazioni aumentano con l'aumento del numero di processori. L'aumento della velocità effettiva viene osservato almeno fino a quando il dodicesimo processore è sottoposto a carichi significativi. Tuttavia, tale aumento è soggetto alle prestazioni di altre risorse durante l'aumento del carico sul sistema. Spesso, la configurazione dei server rispetto a dischi e memoria non è completa. È necessario aggiungere ulteriori processori solo nelle seguenti circostanze:

- Se il carico medio su processori attualmente in uso è oltre il 75% di utilizzo. Se l'utilizzo attuale della CPU è inferiore al 75%, l'aggiunta di ulteriori CPU potrebbe non migliorare le prestazioni.
- Se si verifica un aumento delle prestazioni soddisfacente.

Se OpenText eDirectory è configurato con troppi thread, viene sfruttata una notevole quantità di tempo CPU in commutazione di contesto. In questo caso, una riduzione dei thread può migliorare la velocità effettiva.

Sottosistema di memoria

Le prestazioni delle applicazioni server possono risultare notevolmente migliori quando viene aumentata la RAM. La memorizzazione nella cache del database di OpenText eDirectory nel file system o nella cache FLAIM può migliorare le prestazioni relative alle operazioni di ricerca e modifica. Tuttavia, non è possibile memorizzare nella cache l'intero DIB in installazioni di grandi dimensioni. Evitare lo scambio di pagina anche se ciò implica la riduzione delle dimensioni delle cache di blocchi e voci FLAIM. Per ulteriori informazioni sul sottosistema di memoria, utilizzare lo strumento vmstat.

Grazie alla modalità di utilizzo della memoria da parte di OpenText eDirectory, ogni thread del pool di thread utilizza 1 MB di RAM per il proprio stack. Di default, la dimensione della cache FLAIM è impostata su 200 MB.

Diversi moduli caricabili vengono avviati simultaneamente a OpenText eDirectory, ma l'architettura del modulo caricabile di OpenText eDirectory consente di ridurre il footprint di memoria del processo evitando di caricare i moduli non utilizzati (ad esempio, SecretStore, LDAP o eMBox). Inoltre, prodotti quali IDM dispongono di alcuni moduli che sono eseguiti in OpenText eDirectory.

La memoria utilizzata da OpenText eDirectory potrebbe risultare in aumento. Anche se grazie a un processo di OpenText eDirectory viene liberata memoria, questa potrebbe non essere rilasciata al pool libero del sistema in quanto il gestore della memoria utilizzato internamente da OpenText eDirectory tenta di ottimizzare le allocazioni di memoria per il futuro. Questo è uno dei motivi per cui non è consigliata la configurazione dinamica di FLAIM. Utilizzare lo strumento principale per trovare la dimensione di memoria virtuale approssimativa relativa al processo ndsd presente nell'installazione specifica dell'utente.

La memoria massima che può essere allocata a un processo viene limitata in diversi modi. Una certa quantità di RAM viene utilizzata dal sistema operativo e da altri processi nel sistema. Il sistema operativo può imporre limitazioni alla RAM fisica utilizzata da un processo.

Sottosistema di rete

Le installazioni tipiche dispongono di larghezza di banda sufficiente per gestire il carico di picco della rete. Una larghezza di banda adeguata riduce gli errori, i conflitti e i pacchetti abbandonati. Utilizzare lo strumento netstat per determinare le statistiche di rete.

Diversi sistemi operativi forniscono parametri TCP/IP regolabili per l'ottimizzazione di server di rete complessi. Per informazioni, consultare la documentazione dei sistemi operativi.

Se la rete è il collo di bottiglia, è necessario aumentare la larghezza di banda. La configurazione di una rete privata dedicata tra i server per applicazioni e il server di OpenText eDirectory potrebbe anche facilitare la riduzione della congestione di rete.

4 Ottimizzazione dei sottosistemi OpenText eDirectory

In questa sezione vengono fornite le seguenti informazioni:

- “Database FLAIM” a pagina 19
- “Pool di thread” a pagina 21
- “ACL” a pagina 21
- “Replica” a pagina 24
- “Disco allo stato solido (SSD)” a pagina 25
- “Intervallo di aggiornamento di login NMAS” a pagina 26
- “Overhead SSL” a pagina 26
- “Import Convert and Export (ICE)” a pagina 26
- “ldif2dib” a pagina 26
- “Dimensione del pacchetto NCP migliorata” a pagina 26

Database FLAIM

Il ridimensionamento della cache è con ogni probabilità il fattore che incide maggiormente sulle prestazioni complessive di OpenText eDirectory. Maggiore è il numero di elementi (voci e blocchi) che possono essere memorizzati nella cache, migliore sarà il livello complessivo delle prestazioni. La percentuale di volte in cui i blocchi o le voci vengono trovate nella cache viene denominata frequenza di accesso. Una frequenza più elevata implica prestazioni migliori. OpenText iMonitor consente di visualizzare la frequenza di accesso.

La cache di blocchi risulta più utile per le operazioni di aggiornamento. La cache voci è più utile per quelle operazioni che elaborano ricerche basate su ambito per una voce. Tuttavia, sia le ricerche su un unico livello sia quelle eseguite nel sottoalbero utilizzano la cache voci e la cache di blocchi. La cache di blocchi viene utilizzata per recuperare gli indici. Creare il tipo corretto di indici come richiesto. Per ulteriori informazioni, consultare “Scelta degli indici” a pagina 20.

Durante una lettura del disco può verificarsi un errore nella cache di blocchi. Le letture del disco richiedono sempre molte risorse, ma possono essere evitate se un blocco viene recuperato dalla cache del file system.

La quantità di memoria necessaria per memorizzare l'intero database nella cache di blocchi è quasi pari alla dimensione del database sul disco, e la quantità di memoria richiesta per memorizzare l'intero database nella cache voci è pari a circa due/quattro volte la dimensione del database sul disco. Quando in un sistema si dispone di una quantità di memoria inferiore, provare una cache voci più ridotta e una cache di blocchi o del file system molto più grande.

Se le letture sono localizzate per un set di voci nella directory, è necessario aumentare la cache voci fino a ottenere una migliore frequenza di accesso per tale cache.

Se il modello di lettura è completamente casuale e il DIB è significativamente superiore alla quantità di RAM disponibile, è necessario disporre di una cache di blocchi o di una cache del file system maggiore rispetto alla cache voci.

Qualsiasi metodo utilizzato per ottimizzare OpenText eDirectory allo scopo di migliorarne le prestazioni deve essere provato empiricamente. Un rapporto ottimale di voci per la cache di blocchi in ambienti che sono contraddistinti da un elevato numero di ricerche è 2:1. Assicurarsi che rimanga sufficiente memoria per altri processi. Evitare lo scambio di pagina anche se ciò implica la riduzione delle dimensioni della cache FLAIM.

Poiché FLAIM fornisce una memorizzazione nella cache pre-allocata, la memoria allocata alla cache di OpenText eDirectory non viene mai frammentata dal gestore della memoria del sistema operativo nativo.

Scelta degli indici

Gli indici sono destinati a migliorare le prestazioni delle ricerche eseguite su un unico livello e quelle eseguite nel sottoalbero. Anche i gruppi dinamici utilizzano le ricerche eseguite su un unico livello o nel sottoalbero. Gli indici non vengono utilizzati per le ricerche basate su ambito.

Un indice di presenza viene utilizzato principalmente per scopi interni, in quanto non distingue tra valori presenti e non presenti (eliminati). Se le applicazioni eseguono un'interrogazione di ricerca del tipo di presenza, l'indice non viene utilizzato. Per questo motivo, le applicazioni non devono avere indici di presenza creati appositamente per loro.

Le applicazioni possono creare un indice di valore per un attributo, sufficiente per la maggior parte delle ricerche. FLAIM può utilizzare un indice di valore per l'esecuzione di ricerche sia di presenza sia di stringa secondaria sugli attributi.

Un indice di stringa secondaria può rallentare notevolmente gli aggiornamenti eseguiti su un attributo. Il numero di blocchi dell'indice necessario per supportare un indice di stringa secondaria è piuttosto elevato rispetto all'indice di valore. Ciò significa che per memorizzarli nella cache è necessaria una cache di blocchi maggiore. Creare un indice di stringa secondaria solo se necessario. Un indice di valore dovrebbe essere sufficiente per la maggior parte delle ricerche. Tuttavia, se le ricerche delle stringhe secondarie non producono prestazioni accettabili con un indice di valore, è possibile creare un indice di stringa secondaria su tali attributi.

Se un'operazione di ricerca richiede molto tempo per il completamento, nonostante l'indice selezionato, è possibile introdurre un indice di valore più recente su uno degli attributi del filtro di ricerca. Scegliere l'attributo in grado di produrre risultati migliori quando indicizzato.

Ottimizzazione degli aggiornamenti

La cache di blocchi risulta più utile per le operazioni di aggiornamento. Gli indici si trovano anche nella cache di blocchi. Anche se gli indici consentono di velocizzare le ricerche, una quantità eccessiva di indici mantiene il server occupato per la loro manutenzione. Se i valori degli attributi vengono modificati, aggiunti o eliminati, anche gli indici vengono modificati. Durante operazioni di upload di grandi dimensioni, gli indici possono essere disabilitati permettendo un'esecuzione degli upload più veloce.

Se la directory RFL risiede su un disco diverso rispetto alla directory DIB, ciò consente di migliorare le prestazioni.

Un limite accettabile per il tempo di risposta relativo a un'operazione di aggiornamento può essere controllato mediante la `maxdirtycache`. Ad esempio, se un limite accettabile per la risposta del server è di 5 secondi e la velocità di scrittura casuale del disco è di 20 MB al secondo, la `maxdirtycache` deve essere impostata come 20 x 5 = 100 MB. Assicurarsi che la cache di blocchi possa trattenere questi blocchi modificati nella memoria. Per ulteriori informazioni, vedere "Modifica delle impostazioni della cache FLAIM mediante _ndsdb.ini" a pagina 32.

Pool di thread

Di default, il numero massimo di thread che possono essere disponibili nel pool di thread è 256. Questo numero dovrebbe essere sufficiente per la maggior parte delle installazioni. Tuttavia, nelle grandi installazioni, tale numero può essere aumentato a 512 thread. È necessario aumentare il numero di thread del pool nei seguenti casi:

- Se il numero di thread inattivi spesso è zero.
- Se la quantità di tempo media necessaria per un task nella coda Pronti è elevata e in aumento.
- Se il numero di task nella coda Pronti è elevato e in aumento.

Aumentare il numero massimo di thread se le prestazioni del server aumentano. Tale incremento dovrebbe inoltre risultare in un aumento dell'utilizzo della CPU.

Per informazioni sulla visualizzazione delle statistiche del pool di thread, vedere "Viewing the Thread Pools Statistics" (Visualizzazione delle statistiche dei pool dei thread) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guida all'amministrazione di OpenText™ eDirectory CE 24.4 (v9.3)).

ACL

- "Miglioramento delle letture e delle ricerche di OpenText eDirectory" a pagina 21
- "Disabilitazione dei modelli ACL" a pagina 22

Miglioramento delle letture e delle ricerche di OpenText eDirectory

Una ricerca LDAP in OpenText eDirectory restituisce risultati in base al numero di attributi restituiti per un utente (`inetOrgPerson`).

Quando viene creato un oggetto in OpenText eDirectory, gli ACL di default possono essere aggiunti all'oggetto. Ciò dipende dai modelli ACL nella definizione dello schema per l'objectClass a cui tale oggetto appartiene. Ad esempio, nella configurazione di default di `inetOrgPerson`, possono essere presenti fino a sei ACL aggiunti all'oggetto Utente. Quando viene effettuata una richiesta di ricerca LDAP per restituire l'oggetto Utente con tutti gli attributi, questa impiega un tempo leggermente più lungo per restituire l'oggetto al client piuttosto che restituire l'oggetto Utente privo di attributi ACL.

Benché sia possibile disattivare gli ACL di default, gli amministratori potrebbero decidere di non disattivarli poiché sono necessari per un migliore controllo degli accessi. Tuttavia, è possibile migliorare le prestazioni di ricerca evitandone la richiesta oppure contrassegnandoli come attributi filtrati per la lettura. Tali modifiche non interrompono le applicazioni in quanto la maggior parte di esse utilizza privilegi efficaci e non si basa su ACL specifici.

ACL non richiesti: Un attributo ACL non è richiesto da diverse applicazioni. In questo modo, le applicazioni possono essere modificate affinché richiedano attributi specifici per l'applicazione. Ciò consente di migliorare le prestazioni della ricerca LDAP.

Contrassegno di un ACL come filtrato per la lettura: Se un'applicazione non può essere modificata, un amministratore può utilizzare `arf_acl.ldif` per contrassegnare l'attributo ACL come un attributo filtrato per la lettura. Quando l'ACL viene contrassegnato come un attributo filtrato per la lettura, se sono stati richiesti tutti gli attributi il server non restituisce l'attributo sulla voce. Tuttavia, se la ricerca LDAP viene eseguita per restituire gli attributi operativi oppure vengono richiesti in modo specifico gli attributi ACL, viene restituito l'attributo contrassegnato. `rrf_acl.ldif` consente di disattivare il flag filtrato per la lettura di un attributo ACL. Questi LDIF interessano l'attributo ACL sullo schema. In questo modo, solo un utente che disponga dei diritti di supervisore sulla root della struttura ad albero può estenderli.

Di default, un ACL non è contrassegnato come filtrato per la lettura, quindi il vantaggio relativo alle prestazioni ottenute nella restituzione di tutti gli attributi da parte delle richieste non è visibile.

Nella tabella seguente viene illustrata l'ubicazione dei file `arf_acl.ldif` e `rrf_acl.ldif` in piattaforme diverse.

Piattaforma	Ubicazione
Il test su Linux	◆ `/opt/novell/eDirectory/lib64/nds-schema/`
Windows	◆ `<ubicazione_decompressione>\eDirectory\windows\x64\NDSonNT\ndsnt\nds`

Disabilitazione dei modelli ACL

È possibile disabilitare i modelli di elenco di controllo dell'accesso (ACL, Access Control List) per migliorare le prestazioni dell'inoltro di massa. Ciò implica che alcuni ACL vadano perduti. Tuttavia, è possibile risolvere questo problema aggiungendo gli ACL richiesti al file LDIF o applicarli successivamente.

1 Eseguire il comando seguente:

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

L'output di questo comando risulterà analogo al seguente:

```
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 Nell'output segnalato nel passaggio precedente, eliminare le informazioni in grassetto.

3 Salvare l'output revisionato come un file LDIF.

4 Aggiungere le informazioni seguenti al file LDIF appena salvato:

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: (2.16.840.1.113730.3.2.2)

-

add:objectclasses
```

Di conseguenza, il file LDIF deve essere simile al seguente:

```
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 Immettere il seguente comando:

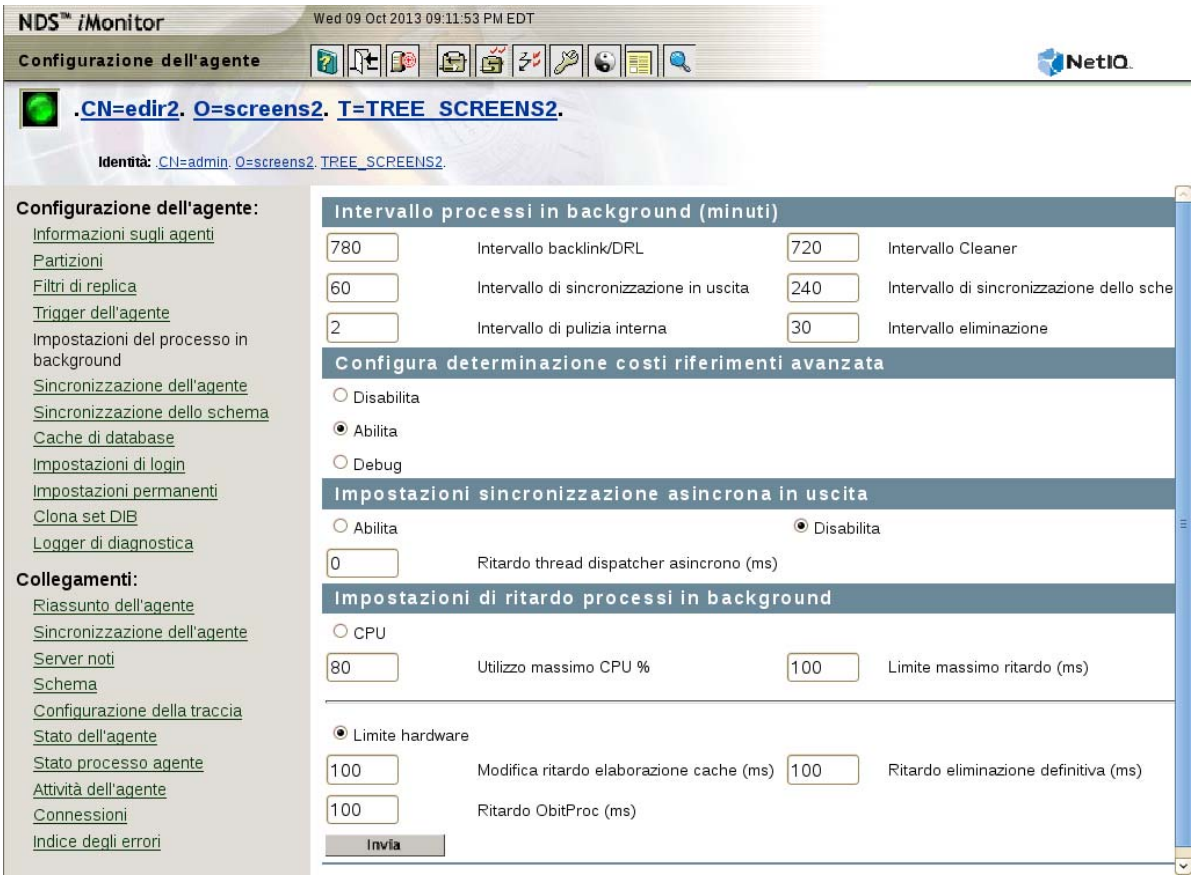
```
ldapmodify -D cn_of_admin -w password -f LDIF_file_name
```

Replica

In questa versione, alcuni processi in background sono stati riprogettati per soddisfare le esigenze di ambienti dinamici di grandi dimensioni. Per ulteriori informazioni, vedere “Managing Background Process” (Gestione dei processi in background) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guida all'amministrazione di OpenText™ eDirectory CE 24.4 (v9.3)).

È consigliato impostare il limite a *5 ms* e di *abilitare* Sincronizzazione in uscita asincrona. Tuttavia, se l'utilizzo della CPU aumenta, aumentare la durata del tempo di inattività. Figura 4-1 mostra i valori impostati per le **impostazioni di ritardo del processo in background**.

Figura 4-1 *Impostazioni del processo in background*



Prove di laboratorio interne sono state eseguite su una configurazione di 10 server con le seguenti impostazioni: Limite rigido - *0 ms*, Sincronizzazione in uscita asincrona - *abilitata* e Ritardo thread dispatcher asincrono - *0 ms*. Le prove hanno dimostrato che la replica è 7 volte più veloce con le impostazioni di default. Durante questa prova, non sono state eseguite altre operazioni client.

Nota: Per sfruttare al meglio i vantaggi delle prestazioni dei sistemi con questi miglioramenti di scalabilità, è necessario utilizzare OpenText eDirectory 9.1 o superiore su tutti i server. Anche se nell'anello di replica sono presenti alcune versioni meno recenti, si riscontra comunque un miglioramento delle prestazioni.

Disco allo stato solido (SSD)

Questa versione supporta SSD Enterprise per operazioni I/O potenziate. Tabella 4-1 a pagina 25 mostra il miglioramento delle prestazioni di riparazione su SSD nella configurazione di prova:

Tabella 4-1 *Prestazioni di riparazione*

Dimensione DIB (GB)	HDD (tempo in minuti)	SSD (tempo in minuti)	% di miglioramento
11	80	53	33.75

Dimensione DIB (GB)	HDD (tempo in minuti)	SSD (tempo in minuti)	% di miglioramento
24	277	169	38,98
34	542	296	45,38
75	1383	618	55,31
98	3171	1023	67,73

Intervallo di aggiornamento di login NMAS

Per ulteriori informazioni, vedere “Using the sasUpdateLoginInfo and sasUpdateLoginTimeInterval Attribute” (Utilizzare l'attributo sasUpdateLoginInfo e sasUpdateLoginTimeInterval) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guida all'amministrazione di OpenText™ eDirectory CE 24.4 (v9.3)).

Overhead SSL

LDAP su SSL consente di aggiungere un ulteriore carico sulla CPU a causa dei relativi requisiti di cifratura. Uno studio di laboratorio relativo alle prestazioni mostra un aumento delle prestazioni pari al 10% a causa dell'overhead di cifratura.

Import Convert and Export (ICE)

L’utility OpenText Import Convert and Export (ICE) utilizza un protocollo di aggiornamento di massa ottimizzato denominato LBURP per eseguire l'upload dei dati in OpenText eDirectory. Questo protocollo è notevolmente più veloce dell'upload dei dati utilizzando un semplice comando `ldapmodify`. Per ulteriori informazioni, vedere Offline Bulkload Utility (Utility per l’inoltro di massa offline) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide (Guida all'amministrazione di OpenText™ eDirectory CE 24.4 (v9.3)).*

ldif2dib

Per l'ottimizzazione delle prestazioni di OpenText eDirectory durante l'upload di massa offline mediante l'utility ldif2dib, consultare Tuning ldif2dib (Ottimizzazione di ldif2dib) nella *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide (Guida all'amministrazione di OpenText™ eDirectory CE 24.4 (v9.3)).*

Dimensione del pacchetto NCP migliorata

Per comunicare tra diversi server, come protocollo di comunicazione OpenText eDirectory utilizza il protocollo NCP (NetWare Core Protocol). Nelle versioni precedenti, la dimensione massima del pacchetto consentita da NCP equivaleva a 64 KB, il che limitava la velocità effettiva del trasferimento dei dati su NCP. In questa versione è stata migliorata la capacità di NCP di gestire pacchetti di dimensioni fino a 1 MB, consentendo a OpenText eDirectory di sincronizzare fino a 1 MB di dati in un unico pacchetto. OpenText eDirectory avvia la sincronizzazione con una dimensione del pacchetto di

64 KB, aumentando tale dimensione in base ai dati che devono essere ancora sincronizzati. In questo modo le prestazioni della replica sono state significativamente potenziate. Se nei due server è in esecuzione la versione 9.2, non è necessario effettuare alcuna ulteriore configurazione per sfruttare al meglio questo miglioramento.

5 Configurazione di OpenText eDirectory

In questa sezione vengono fornite le seguenti informazioni:

- “Configurazione del sottosistema FLAIM” a pagina 29
- “Modifica delle impostazioni della cache FLAIM” a pagina 29

Configurazione del sottosistema FLAIM

Per soddisfare una vasta gamma di configurazioni e installazioni, in OpenText eDirectory sono disponibili due meccanismi mediante i quali controllare il consumo della memoria cache. Questi meccanismi si escludono a vicenda.

- “Limite rigido della cache” a pagina 29
- “Regolazione dinamica del limite” a pagina 29

Limite rigido della cache

È possibile specificare un limite rigido di memoria in uno dei seguenti modi:

- Come un numero fisso di byte.
- Come una percentuale di memoria fisica.
- Come una percentuale di memoria fisica disponibile.

Quando viene specificato un limite rigido utilizzando il secondo o il terzo metodo, tale limite viene sempre convertito in un numero fisso di byte. Ciò significa che per il secondo metodo, il numero di byte rappresenta la percentuale di memoria fisica rilevata all'avvio di OpenText eDirectory. Per il terzo metodo, il numero di byte rappresenta la percentuale di memoria fisica disponibile rilevata all'avvio di OpenText eDirectory.

Regolazione dinamica del limite

Una regolazione dinamica provoca che OpenText eDirectory regoli periodicamente il consumo di memoria in risposta al consumo di memoria variabile relativo ad altri processi. Anche se la regolazione dinamica della memoria funziona correttamente negli scenari tipici, questo meccanismo non è consigliabile per ottenere prestazioni efficaci di OpenText eDirectory su piattaforme Linux a causa delle grandi differenze dei modelli di utilizzo di memoria e degli allocatori di memoria sulle piattaforme Linux.

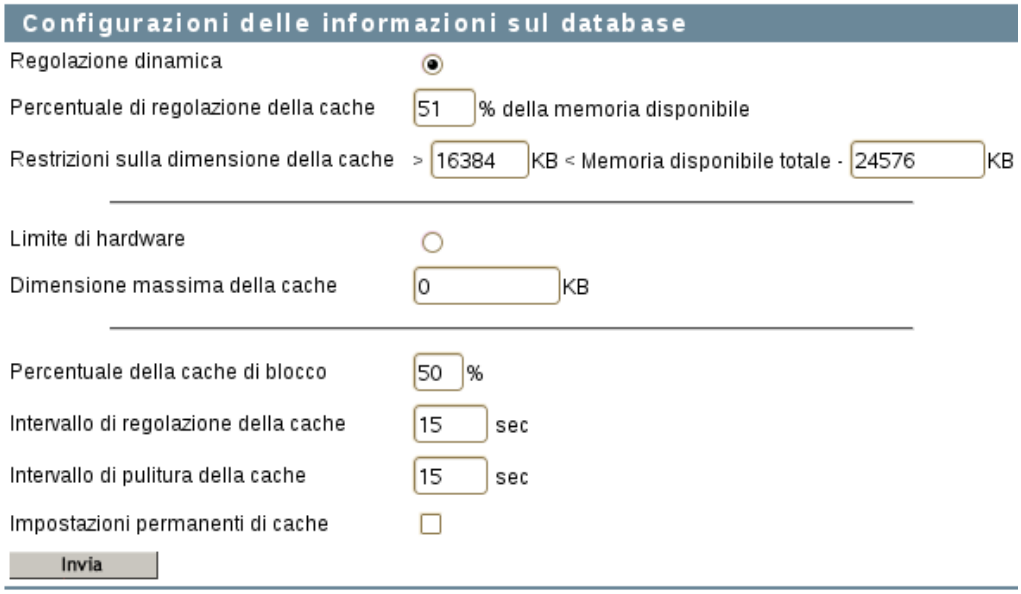
Modifica delle impostazioni della cache FLAIM

- “Modifica delle impostazioni della cache FLAIM mediante OpenText iMonitor” a pagina 30
- “Modifica delle impostazioni della cache FLAIM mediante _ndsdb.ini” a pagina 32

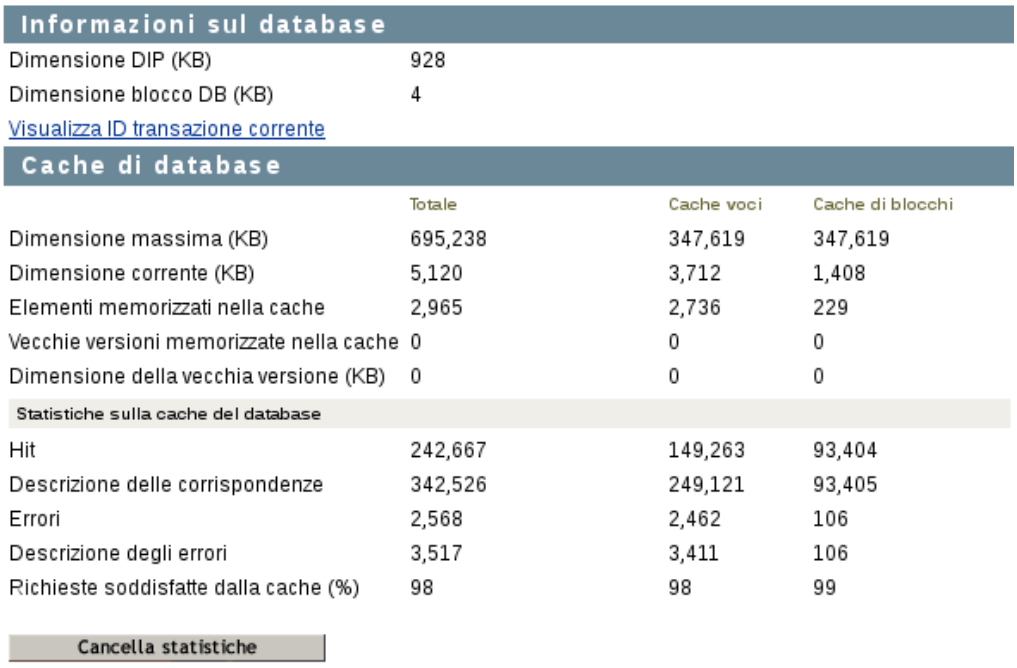
Modifica delle impostazioni della cache FLAIM mediante OpenText iMonitor

È possibile usare OpenText iMonitor per eseguire le seguenti operazioni:

- Visualizzare o modificare le impostazioni della cache.



- Monitorare le statistiche della cache.



Per le informazioni descritte sopra, consultare Cache del database in Configurazione dell'agente di OpenText iMonitor.

Informazioni sulla cache del database	Descrizione
Dimensione massima	La dimensione massima (in KB) che può essere raggiunta dalla cache specificata.
Dimensione attuale	La dimensione attuale (in KB) della cache specificata.
Elementi memorizzati nella cache	Il numero di elementi presenti nella cache specificata.
Versioni precedenti memorizzate nella cache	Il numero di versioni precedenti presenti nella cache specificata. Gli elementi della cache relativi a versioni precedenti sono conservati per mantenere intatta la coerenza delle transazioni di lettura nel database. In altre parole, se un thread si trova in una transazione di lettura e un altro si trova in una transazione di scrittura, le versioni precedenti dei blocchi modificati dallo scrittore vengono conservate per conto del lettore. In questo modo, i risultati del lettore sono garantiti per produrre una visualizzazione coerente durante la vita della relativa transazione anche se durante tale periodo sono state apportate delle modifiche.
Dimensione di versioni precedenti	La dimensione (in KB) degli elementi della versione precedente memorizzati nella cache.
Corrispondenze	Il numero di volte in cui è stato effettuato correttamente l'accesso a un elemento dalla cache specificata.
Consultazioni corrispondenze	Il numero di elementi consultati nella cache prima di eseguire l'accesso corretto a un elemento dalla cache specificata. Il rapporto relativo alla "consultazione corrispondenza per corrispondenza" rappresenta un sistema di misurazione per il livello di efficienza di ricerca nella cache. In genere, il rapporto dovrebbe essere prossimo a 1:1.
Errori	Il numero di volte in cui un elemento non è stato trovato nella cache specificata ed è stato necessario recuperarlo da una cache di livello inferiore o dal disco.
Errore consultazioni	Il numero di elementi consultati nella cache prima di poter stabilire che l'elemento desiderato non è presente nella cache specificata. Il rapporto "errore consultazione per errore" rappresenta un sistema di misurazione per il livello di efficienza di ricerca nella cache. In genere, il rapporto dovrebbe essere prossimo a 1:1.

Modifica delle impostazioni della cache FLAIM mediante _ndsdb.ini

Le impostazioni della cache FLAIM e altre configurazioni FLAIM possono essere eseguite modificando il file `_ndsdb.ini`, residente nella directory DIB. Riavviare OpenText eDirectory una volta modificato il file `_ndsdb.ini`.

È possibile impostare un limite ad aggiustamento dinamico o impostare un limite rigido della cache. Di seguito sono elencate le opzioni relative alla cache. È possibile specificare diverse opzioni, in qualsiasi ordine, separate da virgole. Tutte sono facoltative.

- **DIN o RIGIDO** - Regolazione dinamica di un limite o impostazione di un limite rigido.
- **% : percentuale** - La percentuale di memoria disponibile o fisica da utilizzare.
- **DISP o TOTALE** - La percentuale specifica la memoria disponibile o la memoria fisica totale. È pertinente solo al limite rigido e senza alcun effetto per il limite ad aggiustamento dinamico, in quanto quest'ultimo viene sempre calcolato in base alla memoria fisica disponibile. Di default è DISP.
- **MIN: byte** - Numero minimo di byte.
- **MAX: byte** - Numero massimo di byte.
- **LASCIA: byte** - Numero minimo di byte da lasciare.

  Ad esempio:

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

  ```
  cache=500000000
  ```

- **preallocatecache: true/false** - Questa impostazione implica che OpenText eDirectory allochi previamente la quantità di memoria specificata dal limite rigido della cache.
- **rfldirectory-** È possibile specificare un percorso diverso per i file RFL.
- **cpinterval** - Numero di secondi dopo il quale FLAIM forza un punto di controllo. Il valore di default è 3 minuti.
- **maxdirtycache** - Numero massimo di byte della cache modificata.
- **lowdirtycache** - Numero minimo di byte della cache modificata.
- **blockcachepercent** - La percentuale della cache FLAIM utilizzata per la cache di blocchi.
- **cacheadjustinterval** - Intervallo in secondi per la regolazione dinamica della cache.
- **cachecleanupinterval** - Intervallo in secondi per la pulizia delle versioni precedenti di voci e blocchi dalla cache.