opentext™

OpenText™ eDirectory CE 24.4 (v9.3)

Guía de ajuste

Enero de 2025

Información legal

Copyright 2009 - 2024 Open Text. Reservados todos los derechos.

La única garantía para los productos y servicios de Open Text y sus afiliados y licenciadores ("Open Text") está definida de forma expresa en la declaración de garantía que acompaña a estos productos y servicios. Nada en este documento debe interpretarse como constituyente de una garantía adicional. Open Text no será responsable de errores técnicos o de redacción ni de omisiones presentes en este documento. La información contenida en este documento está sujeta a cambios sin previo aviso.

Tabla de contenido

Acerca de este libro y la biblioteca 5

1 Descripción general 7

Requisitos previos . . . 7

2 Subsistemas de OpenText eDirectory 9

Base de datos FLAIM . . . 9
- Checkpoint . . . 10
- Índices . . . 10
- Registro de operaciones en progresión . . . 11
- Contenedorización de atributos FLAIM . . . 11

Repositorio de hilos . . . 12

3 Análisis de los cuellos de botella del sistema 15

Subsistema de E/S de disco . . . 15
Subsistema de CPU . . . 16
Subsistema de memoria . . . 17
Subsistema de red . . . 17

4 Optimización de los subsistemas de OpenText eDirectory 19

Base de datos FLAIM . . . 19
- Selección de índices . . . 20
- Optimización de las actualizaciones . . . 21

Repositorio de hilos . . . 21
ACL . . . 21
- Mejora de las búsquedas y lecturas de OpenText eDirectory . . . 22
- Inhabilitación de plantillas ACL . . . 22

Réplica . . . 24
Disco de estado sólido (SSD) . . . 25
Intervalo de actualización de entradas NMAS . . . 26
Sobrecargo de SSL . . . 26
Importación, conversión y exportación (ICE) . . . 26
ldif2dib . . . 26
Tamaño de paquetes NCP mejorado . . . 27

5 Configuración de OpenText eDirectory 29

Configuración del subsistema FLAIM . . . 29
- Límite rígido de caché . . . 29
- Ajuste dinámico del límite . . . 29

Modificación de los ajustes de la caché FLAIM . . . 29
- Modificación de los ajustes de la caché FLAIM a través de OpenText iMonitor . . . 30

Modificación de los ajustes de la caché FLAIM a través de _ndsdb.ini 31

Acerca de este libro y la biblioteca

Aquí se describe cómo analizar y configurar el producto OpenText™ eDirectory CE 24.4 (v9.3) para obtener un rendimiento superior en todas las distribuciones.

Para obtener la versión más reciente de la *Guía de ajuste de OpenText™ eDirectory* , consulte el sitio Web de documentación en línea de OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/).

A quién va dirigida

Esta guía está dirigida a administradores de red.

Otra información de la biblioteca

La biblioteca ofrece los siguientes recursos informativos:

Guía de administración de XDASv2

Describe cómo configurar y usar XDASv2 para auditar OpenText eDirectory y OpenText Identity Manager.

Guía de instalación

Describe cómo instalar OpenText eDirectory. Está dirigida a los administradores de red.

Guía de administración

Describe cómo gestionar y configurar OpenText eDirectory.

Guía de resolución de problemas

Describe cómo solucionar problemas en OpenText eDirectory.

Guía de novedades

Describe las nuevas funciones de OpenText eDirectory.

Estas guías están disponibles en el sitio Web de documentación de OpenText eDirectory (https://www.netiq.com/documentation/edirectory-93/).

Para obtener información sobre la utilidad de gestión de eDirectory, consulte la OpenText Identity Console Administration Guide (https://www.netiq.com/documentation/identity-console/) (Guía de administración de OpenText Identity Console).

1 Descripción general

OpenText eDirectory 9.3 es una solución de servicios de directorio conforme con los estándares, interplataforma, muy ampliable, tolerante a fallos y de alto rendimiento. En esta guía, se proporciona información acerca de cómo ajustar el entorno de OpenText eDirectory para mejorar el rendimiento.

El ajuste para aumentar el rendimiento es una actividad compleja. Requiere conocimientos de OpenText eDirectory y de los subsistemas del sistema operativo. Implica la supervisión del sistema para identificar los cuellos de botella y solucionarlos de uno en uno. Muchas veces los recursos son limitados y el ajuste se limita a OpenText eDirectory y al sistema operativo.

En esta guía, lea la sección Requisitos previos antes de llevar a cabo cualquier tipo de ajuste y después proceda con el resto de secciones. En el capítulo Subsistemas de OpenText eDirectory, se describen los subsistemas principales que influyen en el rendimiento de OpenText eDirectory. En el capítulo Análisis de los cuellos de botella del sistema, se describen diversos recursos del sistema y su influencia en el rendimiento de OpenText eDirectory. En el capítulo Optimización de los subsistemas de OpenText eDirectory, se describe cómo analizar y configurar OpenText eDirectory en diversas condiciones y entornos. Por último, en el capítulo Configuración de OpenText eDirectory, se describe cómo configurar diversos parámetros ajustables.

Requisitos previos

Asegúrese de que se cumplen los siguientes requisitos previos generales antes de intentar configurar el sistema para mejorar su rendimiento:

- Un buen diseño de árbol de OpenText eDirectory puede mejorar el rendimiento de esta solución. Pueden ser aplicables las siguientes consideraciones:
  - Las aplicaciones leen toda la información localmente en el servidor sin necesidad de encadenar las solicitudes.
  - OpenText eDirectory controla automáticamente las referencias a objetos de forma eficiente. Si es posible, los objetos de un servidor no deben hacer referencia a objetos que no son locales en dicho servidor, ya que el mantenimiento de las referencias a objetos no locales puede requerir más tiempo. Si existen dichas referencias, se mantendrán los enlaces en segundo plano. Esto es muy difícil en grandes implantaciones.
  - Si necesita un grupo con 10 000 miembros o más, se recomienda el uso de grupos dinámicos. Esto permite evitar el sobrecargo asociado al mantenimiento de las referencias para muchas personas. Elija la configuración del grupo dinámico con cuidado, porque el uso de varios grupos dinámicos con criterios de búsqueda inadecuados podría sobrecargar el servidor y reducir su rendimiento general. Si una operación de búsqueda tarda mucho tiempo en completarse, es posible que el índice elegido sea ineficaz. Reduzca al mínimo el uso de grupos normales (estáticos), ya que pueden aumentar la cantidad de recorrido de árbol en la entrada.

- Utilice las listas ACL de forma eficaz. Por ejemplo, utilice el [este] Trustee y asígnelo en el nivel de contenedor, en lugar de utilizar una plantilla ACL que se asigne derechos a sí misma. Cuantas menos listas ACL use, mejor será el rendimiento. Para obtener más información sobre las ACL, consulte "OpenText eDirectory Rights" (Derechos de OpenText eDirectory) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).
- Distribuya la carga en varios servidores de réplica.
- Aunque un buen diseño de árbol minimiza la necesidad de recorrido de árbol, a veces es necesario. Puede consultar "Advanced Referral Costing" en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).
- Si la entrada es lenta, puede inhabilitar las actualizaciones durante la entrada. Existen formas independientes para inhabilitar las actualizaciones durante la entrada para los modos NDS y NMAS (servicio de autenticación modular de NetIQ). Sin embargo, es importante comprender las implicaciones de seguridad (http://www.novell.com/documentation/nmas33/admin/data/bg8dphs.html).

- Ejecute comprobaciones de estado a través de iMonitor. Para obtener más información, consulte "Viewing OpenText eDirectory Server Health" (Visualización del estado del servidor de OpenText eDirectory) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]). Asegúrese de lo siguiente:
  - La hora está sincronizada en todos los servidores de réplica.
  - Los procesos en segundo plano y de sincronización de réplica están en buen estado.

2 Subsistemas de OpenText eDirectory

Esta sección trata sobre los subsistemas de OpenText eDirectory.

- “Base de datos FLAIM” en la página 9
- “Repositorio de hilos” en la página 12

Base de datos FLAIM

OpenText eDirectory utiliza una base de datos FLAIM. FLAIM (Flexible Adaptable Information Manager) se utiliza para procesar información tradicional, compleja y volátil. Es un motor de base de datos muy adaptable que admite varios lectores y un modelo de simultaneidad de un solo escritor. Los lectores no bloquean a los escritores y los escritores no bloquean a los lectores.

Físicamente, FLAIM organiza los datos en bloques. Normalmente, algunos de los bloques se almacenan en la memoria. Estos representan la caché de bloques aleatorios. La caché de bloques lógicos (a veces denominada caché de registros) almacena en caché las entradas lógicas de la base de datos. Las entradas se construyen a partir de los elementos de la caché de bloques aleatorios. FLAIM mantiene tablas de algoritmos hash para ambas cachés. El tamaño de la papelera de algoritmos hash se ajusta periódicamente en función del número de elementos.

Por defecto, OpenText eDirectory utiliza un tamaño de bloque de 4 KB. El tamaño de la caché de bloques aleatorios para el almacenamiento en caché del archivo DIB completo es igual que el tamaño del archivo DIB, y el tamaño necesario para la caché de bloques lógicos es entre dos y cuatro veces el tamaño del archivo DIB.

Al recuperar una entrada, FLAIM busca primero la entrada en la caché de bloques lógicos. Si la entrada existe, no es necesaria la lectura desde la caché de bloques aleatorios. Al recuperar un bloque del disco, FLAIM busca primero el bloque en la caché. Si el bloque existe, no es necesaria la operación de lectura de disco.

Cuando se añade o se modifica una entrada, los bloques correspondientes a dicha entrada no se añaden al disco directamente, por lo que puede que el disco y la memoria no estén sincronizados. Sin embargo, las actualizaciones realizadas en la entrada se registran en el registro de operaciones en progresión (RFL). Los RFL se utilizan para recuperar transacciones tras un fallo del sistema.

LRU (menos usados recientemente) es el algoritmo de reemplazo que se utiliza para reemplazar elementos de la caché.

- “Checkpoint” en la página 10
- “Índices” en la página 10
- “Registro de operaciones en progresión” en la página 11
- “Contenedorización de atributos FLAIM” en la página 11

Checkpoint

Un punto de verificación pone la versión de la base de datos disponible en el disco en el mismo estado coherente que la base de datos de la memoria (caché). FLAIM puede aplicar un punto de verificación durante la actividad de actualización mínima de la base de datos. Se ejecuta cada segundo y escribe en el disco los bloques modificados (caché modificada). Los bloques que se modificaron en la caché pero aún no se han escrito en el disco se denominan “bloques modificados”. FLAIM adquiere un bloqueo en la base de datos y realiza la máxima cantidad de trabajo posible hasta que el punto de verificación se completa o hay otro hilo esperando para actualizar la base de datos. Para evitar que la base de datos en disco no esté sincronizada, existen varias situaciones en las que se fuerza un punto de verificación aunque haya hilos esperando para actualizar la base de datos:

- Si el hilo de punto de verificación no puede completar un punto de verificación dentro de un intervalo de tiempo especificado (el valor por defecto es 3 minutos), se fuerza y se borra la caché modificada.
- Si el tamaño de la caché usada es mayor que el de `maxdirtycache` (si se ha definido), se fuerza un punto de verificación para reducir el tamaño de la caché modificada a `mindirtycache` (si se ha definido) o para ponerlo a cero.

Índices

Un índice es un conjunto de claves organizadas de forma que se acelere considerablemente la búsqueda de una clave dada en el índice. Las claves de índice se construyen extrayendo el contenido de uno o más campos (atributos) de las entradas. Los índices se mantienen en la caché de bloques aleatorios. Los cambios realizados en los atributos indexados requieren cambios en los bloques de índice.

OpenText eDirectory define un conjunto de índices por defecto para los atributos (campos) del sistema. Los atributos del sistema como `parentID` y `ancestorID` se utilizan para búsquedas de un nivel y de subárbol. Estos índices no se pueden suspender ni eliminar. El directorio los utiliza internamente. Los índices por defecto se definen para atributos como CN, `apellidos`, `nombre`, etc. Los índices pueden ser de tipo presencia, valor y subcadena. Estos índices se pueden suspender. Cuando se eliminan, se vuelven a crear automáticamente.

Puede utilizar OpenText Identity Console o la utilidad LDAP (protocolo ligero de acceso a directorios) ndsindex para crear índices. Los índices (http://www.netiq.com/documentation/edir83/) son específicos del servidor.

Si habilita la etiqueta Gestor de almacenamiento (`StrMan`) en DSTrace (ndstrace), puede ver el índice seleccionado para las consultas de búsqueda.

El ejemplo siguiente corresponde a un registro de DSTrace para una búsqueda de subárbol con `"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
```

El ejemplo siguiente corresponde a un registro de DSTrace para una búsqueda de subárbol con `"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
```

Para mejorar el rendimiento de la búsqueda con la ordenación del servidor, use la opción -a para anteponer el atributo `AncestorID` a la lista de atributos transferidos al crear un nuevo índice.

En la tabla siguiente, se describe la tendencia a aumentar el tamaño de DIB al crear índices simples/compuestos para un determinado número de objetos. Le recomendamos que consulte estos datos de prueba al crear índices simples/compuestos:

Número de objetos	Aumento del tamaño de DIB con un índice simple que contiene atributos de 20 caracteres	Aumento del tamaño de DIB con un índice simple que contiene atributos de 200 caracteres	Aumento del tamaño de DIB con un índice compuesto que contiene atributos de 20 caracteres
23K	24 KB	3 MB	7 MB
47K	50 KB	6 MB	15 MB
0,5 millones	4 MB	55 MB	100 MB
1 millón	9 MB	111 MB	210 MB

Registro de operaciones en progresión

FLAIM registra las operaciones de cada transacción de actualización en un archivo de registro de operaciones en progresión (RFL). Los RFL se utilizan para recuperar transacciones tras un fallo del sistema o al restaurar desde una copia de seguridad. El archivo RFL se trunca al completar cada punto de verificación, a menos que se active en (`rflkeepfiles`) mediante el uso de una copia de seguridad continua con la DB encendida (http://www.netiq.com/documentation/edir93/).

Contenedorización de atributos FLAIM

Para garantizar la utilización óptima de la caché de entrada y el rendimiento mejorado de las operaciones de búsqueda de atributos, FLAIM almacena los atributos que tienen valores más altos o un mayor número de valores en una ubicación separada, concretamente el contenedor de atributos. Por defecto, los atributos se transferirán automáticamente al contenedor cuando el atributo:

- tenga más de 25 valores.
- tiene un valor mayor que 2048 bytes.

Para inhabilitar la contenedorización automática de atributos, añada `disablemovetoattrcontainer =1` al archivo `_ndsdb.ini` y reinicie OpenText eDirectory.

OpenText eDirectory le proporciona flexibilidad para programar el traslado de atributos. Primero vea qué atributos están listos para su traslado y, después, programe el traslado según sus necesidades.

Para ver el número de atributos que están listos para su traslado a los contenedores de atributos, ejecute el comando `ndscheck`. Para ver los detalles de los atributos, utilice el atributo **dsContainerReadyAttrs** de iMonitor en los objetos de servidor Pseudo de la **configuración de agente**. El usuario también puede encontrar esos atributos que aparecen marcados para la indexación en el estado del agente.

Puede iniciar la contenedorización de atributos utilizando la opción Reparar objeto único de `ndsrepair` para el objeto Pseudoservidor. Para contenedorizar un atributo, emplee el comando `ndsrepair` con el nuevo parámetro avanzado `-am` seguido del nombre del atributo, como se muestra a continuación:

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

Después de mover un atributo al contenedor de atributos, OpenText eDirectory crea un índice del sistema con el nombre del atributo. Cuando se contenedoriza un atributo, no es posible devolverlo al contenedor original.

Nota: Si un atributo tiene un valor superior a 2048 bytes, se seguirá realizando la contenedorización, pero OpenText eDirectory no crea ningún índice del sistema.

Repositorio de hilos

OpenText eDirectory presenta varios hilos por motivos de rendimiento. Cuando se usan varios hilos y el sistema está ocupado, se crean más hilos para controlar la carga y algunos se terminan para evitar un sobrecargo adicional. Resulta ineficiente y costoso crear y destruir hilos con frecuencia. En lugar de generar nuevos hilos y destruirlos para cada tarea, se inician varios hilos y se colocan en un repositorio. El sistema asigna los hilos del repositorio de hilos a varias tareas, según sea necesario. Las tareas se colocan en dos tipos de colas:

- Las tareas que necesitan una programación inmediata se colocan en la cola de hilos preparados.
- Las tareas que necesitan una programación más tarde se colocan en la cola de hilos en espera.

No todos los módulos utilizan el repositorio de hilos. El número real de hilos del proceso es mayor que el número disponible en el repositorio de hilos. Por ejemplo, FLAIM gestiona sus hilos en segundo plano por separado.

Al ejecutar el comando `ndstrace -c threads`, se devuelven las siguientes estadísticas del repositorio de hilos:

- Número total de hilos que se generan, se terminan y están inactivos.
- Número total de hilos de trabajo actuales y número máximo de hilos de trabajo.
- Número de tareas y número máximo de tareas en la cola de hilos preparados.
- Número mínimo, máximo y medio de microsegundos que pasan en la cola de hilos preparados.
- Número actual y máximo de tareas en la cola de hilos en espera.

Ejemplo de un repositorio de hilos de muestra:

```
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
```

Hay ciertos parámetros de repositorio de hilos:

- **n4u.server.max-threads:** Número máximo de hilos que pueden estar disponibles en el repositorio.
- **n4u.server.idle-threads:** Número máximo de hilos inactivos que pueden estar disponibles en el repositorio.
- **n4u.server.start-threads:** Número de hilos iniciados.

Ejecute los comandos `ndsconfig get` y `ndsconfig set` para obtener y establecer el tamaño del repositorio de hilos.

3 Análisis de los cuellos de botella del sistema

Son varios los recursos del sistema que influyen en el rendimiento de OpenText eDirectory. La actualización a la versión más reciente del sistema operativo también mejora el rendimiento.

- “Subsistema de E/S de disco” en la página 15
- “Subsistema de CPU” en la página 16
- “Subsistema de memoria” en la página 17
- “Subsistema de red” en la página 17

Subsistema de E/S de disco

El subsistema de disco es el cuello de botella más habitual. Las operaciones de E/S tardan un tiempo relativamente largo con las colas más largas, lo cual resulta en un uso elevado del disco y ciclos de CPU inactivos. Utilice la herramienta iostat durante las cargas máximas esperadas para determinar los indicadores de tiempo medio de respuesta.

Las operaciones de lectura, escritura y actualización de disco pueden ser secuenciales o aleatorias. Las actualizaciones y lecturas aleatorias constituyen el patrón de acceso más común en las implantaciones de OpenText eDirectory.

Algunas soluciones para cargas de trabajo aleatorias:

- Aumentar la memoria RAM. Esto permite el almacenamiento en caché de datos usados con frecuencia o de datos de lectura anticipada en la capa del sistema de archivos. También permite el almacenamiento en caché del archivo DIB dentro del subsistema FLAIM.
- Usar volúmenes dedicados para el archivo DIB. Mejora el rendimiento del sistema de archivos en volúmenes creados más cerca del eje. Utilizar volúmenes dedicados para registros RFL y de otro tipo.
- A medida que los discos desarrollan una latencia cada vez mayor durante un período de tiempo debido a la fragmentación, deberían desfragmentarse.
- Añadir unidades de disco independientes para el RFL de FLAIM. Este tipo de registro puede realizarse en discos de alta velocidad.
- Utilizar un entorno RAID 10(1+0) con más unidades de disco.

Los archivos creados por OpenText eDirectory pueden aumentar hasta los 4 GB. Los sistemas de archivos optimizados para la gestión de archivos de gran tamaño funcionan de forma eficaz con OpenText eDirectory.

- En Solaris™, el sistema de archivos Veritas* VxFS es un sistema de archivos basado en extent donde los metadatos del sistema de archivos están optimizados para archivos de gran tamaño. El sistema de archivos UFS se basa en bloques de forma indirecta, y los metadatos del sistema de archivos se almacenan en un número mayor de bloques. Incluso pueden estar dispersos para archivos grandes, lo que hace que UFS sea más lento con archivos de mayor tamaño.
- En Linux™, el sistema de archivos Reiser es un sistema rápido con respaldo de transacciones y que funciona mejor que el sistema de archivos ext3 en conjuntos de archivos DIB grandes. Sin embargo, se sabe que el modo con respaldo de transacciones de ext3 ofrece el mismo rendimiento que el sistema de archivos Reiser, aunque el modo por defecto proporciona una mayor coherencia de datos. XFS es un sistema de archivos con respaldo de transacciones de alto rendimiento capaz de manipular archivos de gran tamaño y ofrecer transferencias de datos sin problemas. OpenText eDirectory 9.3 es compatible con plataformas SLES 15 de 32 y 64 bits que utilicen el sistema de archivos XFS.

FLAIM admite un tamaño de bloque de 4 KB y 8 KB. El valor por defecto es 4 KB. Este tamaño coincide con el del bloque por defecto en Linux (`tune2fs -l device`). Sin embargo, en Solaris, el sistema de archivos UFS se crea con un tamaño de bloque por defecto de 8 KB (`df -g mountpoint`). Si el tamaño del bloque FLAIM es menor que el tamaño de bloque del sistema de archivos, pueden darse escrituras de bloques parciales. Si el tamaño del bloque de la base de datos es mayor que el del bloque del sistema de archivos, las lecturas y escrituras de bloques individuales se dividen en una serie de operaciones de E/S físicas distintas. Por lo tanto, siempre debe usar el mismo tamaño para el bloque FLAIM que para el bloque del sistema de archivos.

Los tamaños de bloque solo se pueden controlar durante la creación del archivo DIB. Añada una línea "blocksize=8192" a `_ndsdb.ini` para crear el archivo DIB con un tamaño de bloque de 8 KB.

La elección del tamaño de bloque adecuado depende Del tamaño medio del registro FLAIM en sus implantaciones. Es necesario realizar pruebas empíricas en el conjunto adecuado de datos de prueba para determinar cuál es el mejor tamaño de bloque para su implantación.

Subsistema de CPU

OpenText eDirectory se basa en una arquitectura altamente ampliable. El rendimiento aumenta a medida que se incrementa el número de procesadores. Con cargas elevadas, se observa un mayor rendimiento al menos hasta la adición del duodécimo procesador. Sin embargo, este aumento depende del rendimiento de otros recursos durante la creciente carga del sistema. A menudo los servidores disponen de pocos discos y memoria. Debe añadir más procesadores solo en las siguientes circunstancias:

- Si la carga media de los procesadores usados actualmente supera el 75 % de utilización. Si la utilización de la CPU actual es inferior al 75 %, la adición de más CPU no mejorará el rendimiento.
- Si produce un aumento satisfactorio del rendimiento.

Si OpenText eDirectory se ha configurado con demasiados hilos, se invierte una cantidad de tiempo de CPU considerable en la conmutación de contexto. En este caso, una disminución de los hilos puede mejorar el rendimiento.

Subsistema de memoria

El rendimiento de las aplicaciones de servidor puede mejorar bastante cuando se incrementa la memoria RAM. El almacenamiento en caché de la base de datos de OpenText eDirectory en el sistema de archivos o en la caché FLAIM puede dar lugar a mejoras en las operaciones de búsqueda y modificación. Sin embargo, no se puede almacenar en caché el archivo DIB completo en grandes implantaciones. Evite el intercambio de páginas, aunque esto implique reducir los tamaños de la caché de bloques aleatorios y de entrada FLAIM. Utilice la herramienta vmstat para obtener más información sobre el subsistema de memoria.

OpenText eDirectory utiliza memoria, de modo que cada hilo del repositorio de hilos utiliza 1 MB de memoria RAM para su pila. Por defecto, el tamaño de la caché FLAIM está establecido en 200 MB.

Cuando se inicia OpenText eDirectory, se inician también varios módulos cargables, pero la arquitectura de módulo cargable de OpenText eDirectory le permite reducir el consumo de memoria del proceso al evitar la carga de los módulos no utilizados (por ejemplo, SecretStore, LDAP o eMBox). Además, algunos productos como IDM tienen módulos que se ejecutan dentro de OpenText eDirectory.

Puede parecer que OpenText eDirectory aumenta el consumo de memoria. Aunque se libera memoria mediante un proceso de OpenText eDirectory, es posible que esta no se envíe al repositorio libre del sistema porque el administrador de memoria que OpenText eDirectory utiliza internamente intenta optimizar las asignaciones de memoria para el futuro. Esta es una de las razones por las que no se recomienda la configuración dinámica de FLAIM. Utilice la herramienta Top para determinar el tamaño aproximado de la memoria virtual del proceso ndsd en su implantación.

La cantidad de memoria máxima que se puede asignar a un proceso está limitada de varias maneras. El sistema operativo y otros procesos del sistema utilizan una cantidad determinada de memoria RAM. El sistema operativo puede limitar la memoria RAM física que utiliza un proceso.

Subsistema de red

Las implantaciones típicas tienen ancho de banda suficiente para controlar el pico de carga de red. Un ancho de banda adecuado reduce el número de errores, conflictos y paquetes perdidos. Utilice la herramienta netstat para determinar las estadísticas de la red.

Varios sistemas operativos proporcionan parámetros de TCP/IP ajustables para la optimización de servidores que hacen un uso intensivo de la red. Para obtener información, consulte la documentación de los sistemas operativos.

Si la red es el cuello de botella, debe aumentar el ancho de banda. La configuración de una red privada dedicada entre los servidores de aplicaciones y el servidor de OpenText eDirectory también podría ayudar a reducir la congestión de la red.

4 Optimización de los subsistemas de OpenText eDirectory

En esta sección se incluye la información siguiente:

- “Base de datos FLAIM” en la página 19
- “Repositorio de hilos” en la página 21
- “ACL” en la página 21
- “Réplica” en la página 24
- “Disco de estado sólido (SSD)” en la página 25
- “Intervalo de actualización de entradas NMAS” en la página 26
- “Sobrecargo de SSL” en la página 26
- “Importación, conversión y exportación (ICE)” en la página 26
- “ldif2dib” en la página 26
- “Tamaño de paquetes NCP mejorado” en la página 27

Base de datos FLAIM

El tamaño de la caché es posiblemente el factor que más afecta al rendimiento general de OpenText eDirectory. Cuanto mayor sea el número de elementos (bloques y entradas) que pueden almacenarse en caché, mejor será el rendimiento general. El porcentaje de veces que se encuentran los bloques o las entradas en la caché se conoce como la proporción de impacto. Una proporción de impacto elevada da como resultado un mejor rendimiento. OpenText iMonitor puede utilizarse para ver la proporción de impacto.

La caché de bloques aleatorios es más útil para las operaciones de actualización. La caché de entrada es muy útil para las operaciones que realizan una búsqueda de base para una entrada. Sin embargo, las búsquedas de un nivel y de subárbol utilizan tanto la caché de bloques lógicos como la caché de bloques aleatorios. La caché de bloques aleatorios se utiliza para recuperar los índices. Cree el tipo correcto de índices que sea necesario. Para obtener más información, consulte la sección “Selección de índices” en la página 20.

Un fallo en la caché de bloques aleatorios puede provocar una operación de lectura de disco. Las lecturas de disco siempre resultan caras, pero pueden evitarse si se recupera un bloque de la caché del sistema de archivos.

La cantidad de memoria necesaria para almacenar la base de datos completa en la caché de bloques aleatorios se corresponde casi con el tamaño de la base de datos del disco, y la cantidad de memoria necesaria para almacenar la base de datos completa en la caché de bloques lógicos es entre dos y cuatro veces el tamaño base de datos del disco. Si tiene menos memoria en un sistema, pruebe con una caché de bloques lógicos más pequeña y una caché del sistema de archivos o de bloques aleatorios mucho mayor.

Si las lecturas se reducen a un conjunto de entradas del directorio, aumente la caché de bloques lógicos siempre que el resultado sea una mayor proporción de impacto de la caché de bloques lógicos.

Si el patrón de lectura es completamente aleatorio y el archivo DIB es mucho mayor que la memoria RAM disponible, debe tener una caché de bloques aleatorios o del sistema de archivos más grande que la caché de bloques lógicos.

Cualquier método que utilice para ajustar OpenText eDirectory con el fin de mejorar el rendimiento requiere pruebas empíricas. Una buena proporción entre la caché de bloques lógicos y la caché de bloques aleatorios para entornos con muchas búsquedas sería de 2:1. Asegúrese de que queda suficiente memoria para otros procesos. Evite el intercambio de páginas, aunque esto implique reducir los tamaños de la caché FLAIM.

FLAIM proporciona almacenamiento en caché preasignado, por lo que la memoria asignada a la caché de OpenText eDirectory nunca es fragmentada por el administrador de memoria del sistema operativo nativo.

Selección de índices

Los índices están diseñados para mejorar el rendimiento de las búsquedas de un nivel o de subárbol. Los grupos dinámicos también utilizan búsquedas de un nivel o de subárbol. Los índices no se utilizan para búsquedas de base.

Los índices de presencia no diferencian entre valores presentes y no presentes (eliminados), por lo que se utilizan principalmente para fines internos. Si las aplicaciones ejecutan una consulta de búsqueda de tipo de presencia, este índice no se utiliza nunca, de modo que no se deberían crear índices de presencia para las aplicaciones.

Las aplicaciones pueden crear un índice de valor para un atributo, que es suficiente para la mayoría de las búsquedas. FLAIM puede utilizar un índice de valor para llevar a cabo tanto búsquedas de presencia como búsquedas de subcadena en los atributos.

Un índice de subcadena puede ralentizar considerablemente las actualizaciones realizadas en un atributo. El número de bloques de índice necesario para admitir un índice de subcadena es bastante grande en comparación con el índice de valor. Esto significa que se requiere más caché de bloques aleatorios para almacenarlos en caché. Cree un índice de subcadena solo cuando sea necesario. Un índice de valor debería ser suficiente para la mayoría de las búsquedas. Sin embargo, si las búsquedas de subcadenas no producen un rendimiento aceptable con un índice de valor, puede crear un índice de subcadena en dichos atributos.

Si una operación de búsqueda tarda mucho tiempo en completarse a pesar del índice seleccionado, puede introducir un índice de valor más reciente en uno de los atributos del filtro de búsqueda. Seleccione el atributo que produzca los mejores resultados al indexarlo.

Optimización de las actualizaciones

La caché de bloques aleatorios es más útil para las operaciones de actualización. Los índices también se almacenan en la caché de bloques aleatorios. Aunque los índices agilizan las búsquedas, el hecho de tener demasiados índices ocupa al servidor con su mantenimiento. Los índices se modifican al modificar, añadir o eliminar valores de atributo. Durante las operaciones de carga de grandes volúmenes, los índices se pueden inhabilitar para que la carga sea más rápida.

Tener el directorio RFL en un disco diferente al directorio DIB mejora el rendimiento.

Un límite aceptable para el tiempo de respuesta para una operación de actualización puede controlarse mediante el uso de `maxdirtycache`. Por ejemplo, si 5 segundos es un límite aceptable para la respuesta del servidor y la velocidad de escritura de disco aleatoria es de 20 MB por segundo, `maxdirtycache` debe establecerse en 20x5 = 100 MB. Asegúrese de que la caché de bloques aleatorios puede mantener estos bloques modificados en la memoria. Consulte la "Modificación de los ajustes de la caché FLAIM a través de _ndsdb.ini" en la página 31 para obtener más información.

Repositorio de hilos

Por defecto, el número máximo de hilos que pueden estar disponibles en el repositorio de hilos es 256. Este número debería ser suficiente para la mayoría de las implantaciones. Se puede aumentar a 512 hilos en implantaciones de mayor tamaño. Debe aumentar el número de hilos del repositorio en los casos siguientes:

- Si el número de hilos inactivos es cero con frecuencia.
- Si la media de tiempo que una tarea pasa en la cola de hilos preparados es elevada y va en aumento.
- Si el número de tareas que se encuentran en la cola de hilos preparados es elevado y va en aumento.

Si el rendimiento del servidor aumenta, siga aumentando el número máximo de hilos. También debe provocar una mayor utilización de la CPU.

Para obtener información acerca de cómo ver las estadísticas del repositorio de hilos, consulte "Viewing the Thread Pools Statistics" (Visualización de las estadísticas del repositorio de hilos) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).

ACL

- "Mejora de las búsquedas y lecturas de OpenText eDirectory" en la página 22
- "Inhabilitación de plantillas ACL" en la página 22

Mejora de las búsquedas y lecturas de OpenText eDirectory

Los resultados de una búsqueda LDAP en OpenText eDirectory dependen del número de atributos devueltos para un usuario (`inetOrgPerson`).

Cuando se crea un objeto en OpenText eDirectory, podrían añadirse listas ACL al objeto por defecto. Esto depende de las plantillas ACL de la definición de esquema correspondiente a la clase de objeto a la que pertenezca dicho objeto. Por ejemplo, en la configuración por defecto para `inetOrgPerson`, puede haber hasta seis ACL añadidas al objeto de usuario. Cuando se realiza una solicitud de búsqueda LDAP para devolver este objeto de usuario con todos los atributos, se tarda un poco más en devolver este objeto al cliente que en devolver el objeto de usuario sin atributos ACL.

Aunque las ACL por defecto se pueden desactivar, es posible que los administradores no quieran desactivarlas porque son necesarias para un mejor control de acceso. Sin embargo, puede mejorar el rendimiento de la búsqueda si no las solicita o las marca como atributos filtrados para lectura. Estos cambios no interrumpen ninguna aplicación, ya que la mayoría de las aplicaciones utilizan privilegios eficaces y no dependen de ACL específicas.

No solicitar listas ACL: varias aplicaciones no necesitan un atributo ACL, por lo que las aplicaciones pueden modificarse para que soliciten atributos específicos en los que la aplicación esté interesada. Esto da como resultado un mejor rendimiento de la búsqueda LDAP.

Marcar una ACL como filtrada para lectura: Si una aplicación no se puede modificar, un administrador puede utilizar el atributo `arf_acl.ldif` para marcar el atributo ACL como filtrado para lectura. Cuando la ACL se marca como atributo filtrado para lectura, el servidor no devuelve el atributo en la entrada si se solicitan todos los atributos. Sin embargo, si la búsqueda LDAP se realiza para devolver atributos operativos o si la solicitud solicita específicamente atributos ACL, se devuelve el atributo marcado. `rrf_acl.ldif` puede utilizarse para desactivar el indicador filtrado para lectura en un atributo ACL. Estos archivos LDIF afectan al atributo ACL del esquema, por lo que solo puede extenderlos un usuario que tenga derechos de supervisor en la raíz del árbol.

Por defecto, las ACL no están marcadas como filtradas para lectura, por lo que no se aprecia la mejora de rendimiento en las solicitudes de devolución de todos los atributos.

En la tabla siguiente se muestra la ubicación de los archivos `arf_acl.ldif` y `rrf_acl.ldif` en diferentes plataformas.

Plataforma	Ubicación
Linux	◆ `/opt/novell/eDirectory/lib64/nds-schema/`
Windows	◆ `<ubicación_sin_comprimir>\eDirectory\windows\x64\NDSonNT\ndsnt\nds`

Inhabilitación de plantillas ACL

Puede inhabilitar las plantillas de lista de control de acceso (ACL) para aumentar el rendimiento de carga masiva. Esto implica que faltarán algunas de las ACL. Sin embargo, puede añadir las ACL necesarias al archivo LDIF o aplicarlas más tarde para solucionar este problema.

1 Ejecute el comando siguiente:

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

La salida de este comando sería la siguiente:

```
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 En la salida indicada en el paso anterior, elimine la información marcada en negrita.

3 Guarde los resultados revisados como un archivo LDIF.

4 Añada la siguiente información en el archivo LDIF recién guardado:

```
dn: cn=schema

changetype: modify

delete: objectclasses

objectclasses: (2.16.840.1.113730.3.2.2)

-

add:objectclasses
```

Ahora su LDIF debe ser similar al siguiente:

```
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 Introduzca el siguiente comando:

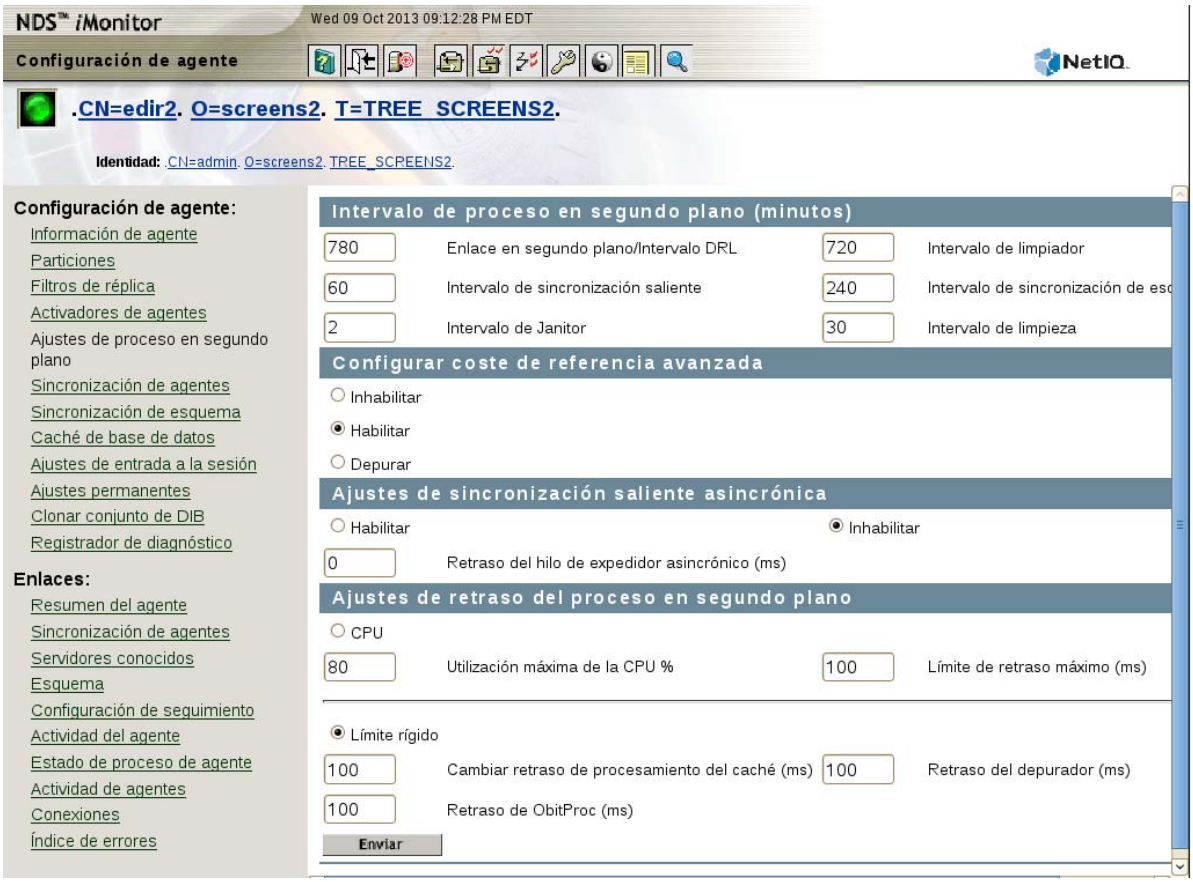
```
ldapmodify -D cn_of_admin -w password -f LDIF_file_name
```

Réplica

En esta versión, se han rediseñado algunos procesos en segundo plano para adaptarlos a entornos grandes y dinámicos. Para obtener más información, consulte “Managing Background Process” (Gestión de procesos en segundo plano) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).

Se recomienda establecer el límite rígido en *5 ms* y *habilitar* la sincronización saliente asincrónica. Sin embargo, si la utilización de la CPU es elevada, aumente la duración del reposo. En la Figura 4-1 se muestran los valores definidos para los **ajustes de demora de procesos en segundo plano**.

Figura 4-1 *Ajustes de proceso en segundo plano*



Se realizaron pruebas de laboratorio interna en una configuración de 10 servidores con los siguientes ajustes: límite rígido de *0 ms*, sincronización saliente asincrónica *habilitada* y demora del hilo expedidor asincrónico de *0 ms*. Las pruebas han demostrado que la replicación es siete veces más rápida que con los ajustes por defecto. Durante esta prueba, no se realizaron otras operaciones de cliente.

Nota: Para obtener los principales beneficios del rendimiento de sus sistemas con estas mejoras de capacidad de ampliación, debe tener OpenText eDirectory 9.1 o una versión posterior en todos los servidores. Aunque haya algunas versiones anteriores del anillo de réplica, se consigue una mejora de rendimiento.

Disco de estado sólido (SSD)

Esta versión admite SSD de empresa para operaciones de E/S mejoradas. En la Tabla 4-1 en la página 26 se muestra la mejora en el rendimiento de la reparación de SSD en nuestra configuración de prueba:

Tabla 4-1 *Rendimiento de la reparación*

Tamaño de DIB (GB)	HDD (tiempo en minutos)	SSD (tiempo en minutos)	% de mejora
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

Intervalo de actualización de entradas NMAS

Para obtener más información, consulte "Using the sasUpdateLoginInfo and sasUpdateLoginTimeInterval Attribute" (Uso de los atributos sasUpdateLoginInfo y sasUpdateLoginTimeInterval) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide* (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).

Sobrecargo de SSL

LDAP sobre SSL añade una carga adicional a la CPU debido a sus requisitos de cifrado. Un estudio de rendimiento de laboratorio muestra un rendimiento superior al 10 % debido al sobrecargo del cifrado.

Importación, conversión y exportación (ICE)

La utilidad de importación, conversión y exportación (ICE) de OpenText utiliza un protocolo de actualización masiva optimizado denominado LBURP para cargar datos en OpenText eDirectory. Este protocolo es significativamente más rápido que cargar datos mediante un sencillo comando `ldapmodify`. Para obtener más información, consulte Offline Bulkload Utility (Utilidad de carga masiva sin conexión) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).*

ldif2dib

Para obtener más información sobre el ajuste del rendimiento de OpenText durante la carga masiva sin conexión con la utilidad ldif2dib, consulte Tuning ldif2dib (Ajuste de ldif2dib) en *OpenText™ eDirectory CE 24.4 (v9.3) Administration Guide (Guía de administración de OpenText™ eDirectory CE 24.4 [v9.3]).*

Tamaño de paquetes NCP mejorado

Para la comunicación en varios servidores, OpenText eDirectory utiliza el protocolo NCP (Netware Core Protocol). En versiones anteriores, NCP permitía un tamaño de paquete máximo de 64 KB, lo cual limitaba el rendimiento máximo al transferir datos a través de NCP. Esta versión mejora la capacidad de NCP para gestionar paquetes de hasta 1 MB para que OpenText eDirectory puede sincronizar hasta 1 MB de datos en un solo paquete. OpenText eDirectory empieza a sincronizar con un tamaño de paquete de 64 KB y aumenta dicho tamaño en función de los datos que falten por sincronizar. Esto mejora considerablemente el rendimiento de las réplicas. Si la versión de ambos servidores es 9.2, no es necesario que configure nada más para aprovechar esta mejora.

5 Configuración de OpenText eDirectory

En esta sección se incluye la información siguiente:

- “Configuración del subsistema FLAIM” en la página 29
- “Modificación de los ajustes de la caché FLAIM” en la página 29

Configuración del subsistema FLAIM

Para abordar una amplia gama de configuraciones y distribuciones, en OpenText eDirectory, se proporcionan dos mecanismos para controlar el consumo de memoria caché. Estos mecanismos son mutuamente excluyentes.

- “Límite rígido de caché” en la página 29
- “Ajuste dinámico del límite” en la página 29

Límite rígido de caché

Puede especificar un límite rígido de memoria de una de las maneras siguientes:

- como un número determinado de bytes;
- como un porcentaje de memoria física;
- como un porcentaje de memoria física disponible.

Cuando se especifica un límite rígido utilizando el segundo o el tercer método, siempre se traduce en un número fijo de bytes. Esto significa que, para el segundo método, el número de bytes es el porcentaje de memoria física detectado al iniciar OpenText eDirectory. En el caso del tercer método, el número de bytes es el porcentaje de memoria física disponible detectado al iniciar OpenText eDirectory.

Ajuste dinámico del límite

Cuando se usa el ajuste dinámico, OpenText eDirectory ajusta el consumo de memoria de forma periódica en respuesta al consumo de memoria variable de otros procesos. El ajuste dinámico de memoria funciona eficazmente en situaciones típicas, pero este mecanismo no se recomienda para un rendimiento óptimo de OpenText eDirectory en plataformas Linux debido a las grandes diferencias en los patrones de uso de memoria y los asignadores de memoria de las plataformas Linux.

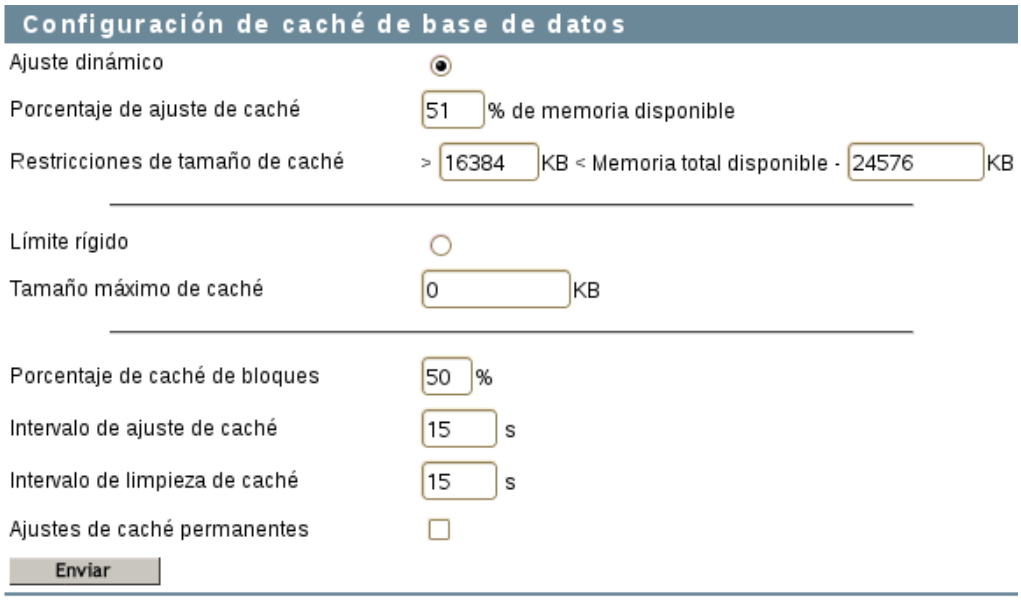
Modificación de los ajustes de la caché FLAIM

- “Modificación de los ajustes de la caché FLAIM a través de OpenText iMonitor” en la página 30
- “Modificación de los ajustes de la caché FLAIM a través de _ndsdb.ini” en la página 31

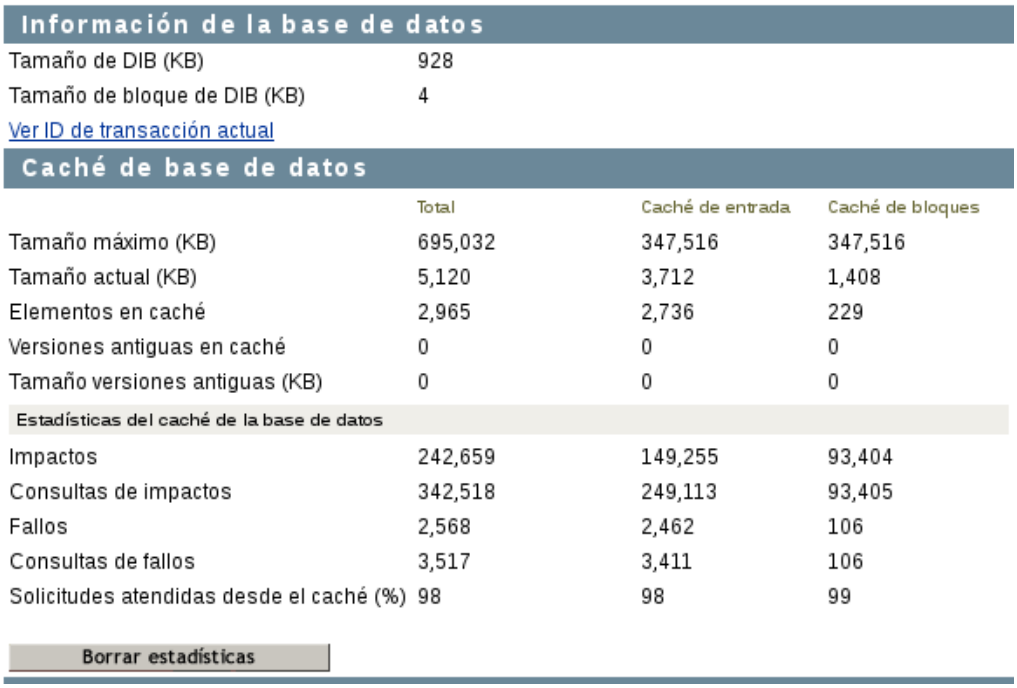
Modificación de los ajustes de la caché FLAIM a través de OpenText iMonitor

Puede utilizar OpenText iMonitor para realizar las operaciones siguientes:

- Ver o cambiar los ajustes de la caché.



- Supervisar las estadísticas de memoria caché.



Consulte la caché de base de datos en la configuración del agente de OpenText iMonitor para obtener la información anterior.

Información de la caché de base de datos	Descripción
Tamaño máximo	Tamaño máximo (en KB) que puede crecer a la memoria caché especificada.
Tamaño actual	Tamaño actual (en KB) de la caché especificada.
Elementos en caché	Número de elementos de la caché especificada.
Versiones anteriores en caché	Número de versiones anteriores en la caché especificada. Las versiones anteriores de los elementos en caché se conservan para preservar la coherencia de las transacciones de lectura en la base de datos. En otras palabras, si un hilo se encuentra en una transacción de lectura y otro en una transacción de escritura, las versiones anteriores de los bloques modificados por el sistema de escritura se mantienen en nombre del lector. Esto se hace para garantizar que los resultados del lector producen una vista coherente durante la transacción, aunque se produzcan modificaciones.
Tamaño de las versiones anteriores	Tamaño (en KB) de los elementos de la versión anterior en caché.
Impactos	Número de veces que se accede a un elemento desde la caché especificada.
Consultas de impacto	Número de elementos consultados en la caché antes de que se acceda a un elemento correctamente desde la caché especificada. La proporción de consultas de impacto e impactos es una medida de la eficacia de la búsqueda en la caché. Normalmente, la proporción debe acercarse a 1:1.
Fallos	Número de veces que no se encontró un elemento en la caché especificada y que se tuvo que obtener de una caché de nivel inferior o del disco.
Consultas de fallos	Número de elementos consultados en la caché antes de determinar que el elemento deseado no se encuentra en la caché especificada. La proporción de consultas de fallos y fallos es una medida de la eficacia de la búsqueda en la caché. Normalmente, la proporción debe acercarse a 1:1.

Modificación de los ajustes de la caché FLAIM a través de _ndsdb.ini

La modificación del archivo `_ndsdb.ini` situado en el directorio DIB permite configurar los ajustes de la caché FLAIM y otras configuraciones de FLAIM. Reinicie OpenText eDirectory cuando se modifique el archivo `_ndsdb.ini`.

Puede definir el límite de ajuste dinámico o el límite rígido de caché. A continuación se enumeran las opciones de caché. Se pueden especificar varias opciones, en cualquier orden, separadas por comas. Todas son opcionales.

- **DYN o HARD:** ajuste dinámico de un límite o de un límite rígido.

- **% : porcentaje** porcentaje de memoria física o disponible para utilizar.
- **AVAIL o TOTAL:** el porcentaje especifica la memoria disponible o el total de memoria física. Solo es aplicable en el caso del límite rígido y se ignora en el límite de ajuste dinámico, ya que el ajuste dinámico de límites se calcula siempre en función de la memoria física disponible. El valor por defecto es AVAIL.
- **MIN: bytes:** número mínimo de bytes.
- **MAX: bytes:** número máximo de bytes.
- **LEAVE: bytes:** número mínimo de bytes que se deben dejar.

  Por ejemplo:

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

  ```
  cache=500000000
  ```

- **preallocatecache: verdadero/falso:** este ajuste hace que OpenText eDirectory asigne previamente la cantidad de memoria especificada por el límite rígido de caché.
- **rfldirectory:** puede especificarse una vía diferente para los archivos RFL.
- **cpinterval:** número de segundos tras los que FLAIM fuerza un punto de verificación. El valor por defecto es de 3 minutos.
- **maxdirtycache:** máximo de bytes de caché modificados.
- **lowdirtycache:** mínimo de bytes de caché modificados.
- **blockcachepercent:** porcentaje de la caché FLAIM utilizado para la caché de bloques aleatorios.
- **cacheadjustinterval:** intervalo en segundos para ajustar dinámicamente la caché.
- **cachecleanupinterval:** intervalo en segundos para la limpieza de versiones anteriores de las entradas y los bloques de la caché.